EVALUASI RISIKO CELAH KEAMANAN MENGGUNAKAN *METODOLOGI OPEN WEB APPLICATION SECURITY PROJECT* (OWASP) PADA APLIKASI *WEB* SISTEM INFORMASI AKADEMIK (SIAKAD) UIN AR-RANIRY

TUGAS AKHIR

Diajukan Oleh:

DARUL FATA
NIM. 180705016
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI
AR-RANIRY
2023 M / 1445 H

EVALUASI RISIKO CELAH KEAMANAN MENGGUNAKAN *METODOLOGI OPEN WEB APPLICATION SECURITY PROJECT* (OWASP) PADA APLIKASI *WEB* SISTEM INFORMASI AKADEMIK (SIAKAD) UIN AR-RANIRY

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri Ar-Raniry Banda Aceh
Sebagai Beban Studi Memperoleh Gelar Sarjana Dalam Ilmu Teknologi Informasi

Oleh:

DARUL FATA
NIM. 180705016
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui untuk Dimunaqasyahkan Oleh:

Pembimbing I,	Pembimbing II,
Malahayati, M.T.	**Mulkan Fadhli, S.T., M.T.**
NIP. 198301272015032003	**NIP. 198811282020121006**

Mengetahui
Ketua Program Studi Teknologi Informasi

Ima Dwitawati, M.B.A
NIP. 198210132014032002

EVALUASI RISIKO CELAH KEAMANAN MENGGUNAKAN *METODOLOGI OPEN WEB APPLICATION SECURITY PROJECT* (OWASP) PADA APLIKASI *WEB* SISTEM INFORMASI AKADEMIK (SIAKAD) UIN AR-RANIRY

TUGAS AKHIR

Telah diuji oleh Panitia Munaqasyah Skripsi
Fakultas Sains dan Teknologi UIN Ar-Raniry dan dinyatakan Lulus
Serta diterima sebagai Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
dalam Ilmu Teknologi Informasi

Pada Hari/Tanggal : Kamis, 27 Juli 2023
9 Muharram 1445 H
di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,

Malahayati, M.T.
NIP. 198301272015032003

Sekretaris,

Mulkan Fadhli, S.T., M.T.
NIP. 198811282020121006

Penguji I,

Hendri Ahmadian, M.I.M.
NIP. 198301042014031002

Penguji II,

Khairan AR, M.Kom.
NIP. 198607042014031001

Mengetahui,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Ar-Raniry Banda Aceh

Dr. Ir. Muhammad Dirhamsyah, M.T.,IPU
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN KARYA ILMIAH/TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	: Darul Fata
NIM	: 180705016
Program Studi	: Teknologi Informasi
Fakultas	: Sains dan Teknologi
Judul Tugas akhir	: Evaluasi Risiko Celah Keamanan Menggunakan Metodologi *Open Web Application Security Project* (OWASP) Pada Aplikasi *Web* Sistem Informasi Akademik (Siakad) UIN Ar-Raniry

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu mempertanggungjawab atas karya ini;

Bila kemudian hari ini ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat mempertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenakan sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh. Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 25 Juli 2023

Yang Menyatakan,

Darul Fata

ABSTRAK

Nama	: Darul Fata
NIM	: 180705016
Program Studi	: Teknologi Informasi
Fakultas	: Sains dan Teknologi (FST)
Judul	: Evaluasi Risiko Celah Keamanan Menggunakan Metodologi *Open Web Application Security Project* (OWASP) Pada Aplikasi *Web* Sistem Informasi Akademik (Siakad) UIN Ar-Raniry
Tanggal Sidang	: 27 Juli 2023
Tebal Tugas Akhir	: 66 Halaman
Pembimbing I	: Malahayati, M.T.
Pembimbing II	: Mulkan Fadhli, S.T., M.T.

Penggunaan aplikasi Sistem Informasi berbasis *web* seperti Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry merupakan faktor penting bagi kemajuan organisasi pendidikan. Namun, sistem informasi berbasis *web* rentan terhadap celah keamanan yang dapat dieksploitasi melalui internet dan dapat berpotensi menyebabkan kerugian. Tujuan penelitian ini adalah mengevaluasi keamanan Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry menggunakan metode *Open Web Application Security Project* (OWASP) dengan alat OWASP Zap yang merupakan scanner otomatis untuk menemukan kerentanan keamanan. Berdasarkan hasil penelitian menunjukkan adanya kerentanan dengan tingkat risiko sedang dan lemah, menandakan potensi risiko keamanan *website* yang masih cukup tinggi. Tindakan mitigasi untuk meningkatkan keamanan SIAKAD mencakup penerapan pesan kesalahan kustom, pengaturan logging yang aman, penyaringan input pengguna, monitoring aktivitas, pengujian keamanan rutin, dan pemasangan pembaruan keamanan. Selain itu, edukasi pengguna tentang praktik keamanan dan implementasi kebijakan keamanan juga menjadi sangat penting. Koneksi *HTTPS* dan pengelolaan *cookie* dengan *Secure Flag* dan *SameSite Attribute* diterapkan untuk melindungi data sensitif, sementara pemantauan *cookie* dan evaluasi rutin laporan kebijakan menjadi bagian penting dari strategi mitigasi keamanan pada sistem informasi akademik.

Kata Kunci: OWASP, Evaluasi Keamanan Sistem Informasi, *Penetration Testing*

KATA PENGANTAR

بِسْـــــــــــــمِ اللهِ الرَّحْمَنِ الرَّحِيْمِ

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan penulisan tugas akhir ini dengan baik. Shalawat dan salam tidak lupa kita sanjung sajikan kepada baginda Nabi besar Muhammad SAW yang telah membawa seluruh umatnya untuk menjadi generasi yang berilmu pengetahuan.

Dengan izin Allah SWT, saya dapat menyelesaikan tugas akhir yang berjudul **"Evaluasi Risiko Celah Keamanan Menggunakan Metodologi *Open Web Application Security Project* (OWASP) Pada Aplikasi *Web* Sistem Informasi Akademik (Siakad) UIN Ar-Raniry"** Dengan harapan penulis bahwa tugas akhir ini dapat memberikan manfaat bagi pihak yang membutuhkan, menambahkan wawasan serta imu pengetahuan.

Penulis menyadari tugas akhir ini tidak dapat diselesaikan dengan baik tanpa bimbingan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada pihak-pihak yang terlibat secara langsung maupun tidak langsung dalam mendukung kelancaran penulisan tugas akhir ini baik berupa dukungan, doa maupun bimbingan yang telah diberikan. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada:

1. Orang tua yang penulis cintai, Bapak Muhammad dan Ibu Indra Hayati yang telah mendo'akan serta memberikan semangat, kasih sayang yang tiada henti kepada penulis serta seluruh keluarga besar yang telah memberikan nasihat, semangat dan motivasi sehingga penulis menyelesaiakan tugas akhir ini.
2. Bapak Dr. Ir. Muhammad Dirhamsyah, M.T., IPU selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.
3. Ibu Ima Dwitawati, M.B.A dan Bapak Khairan AR, M.Kom selaku Ketua dan Sekretaris Program Studi Teknologi Informasi Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

4. Ibu Malahayati, M.T. selaku pembimbing I dan Bapak Mulkan Fadhli, S.T., M.T. selaku pembimbing II yang telah meluangkan waktu, pikiran dan tenaga dalam membimbing penulis demi kesempurnaan tugas akhir ini. Terima kasih banyak penulis ucapkan.
5. Bapak Mulkan Fadhli, S.T., M.T. Selaku Penasehat Akademik (PA) yang telah membimbing saya dalam proses perkuliahan dalam menempuh pendidikan di Program Studi Teknologi Informasi. Terima kasih banyak telah memberi nesehat dan saran selama ini kepada penulis.
6. Seluruh dosen yang mengajar pada Program Studi Teknologi Informasi yang telah memberikan ilmu pengetahuan yang sangat berguna bagi penulis selama proses belajar mengajar.
7. Ibu Cut Ida Rahmadiana S.Si. Selaku staf prodi Teknologi Informasi yang telah membantu penulis dalam hal administrasi selama menempuh pendidikan di Program Studi Teknologi Informasi.
8. Sahabat-sahabat seperjuangan Teknologi Informasi, berbagi semangat dan suka duka dalam penyelesaian tugas akhir ini. Terima kasih banyak penulis ucapkan kepada Rizki Mardhatillah, Nisa Afdhilla, Muhammad Firdaus, Muhammad Ridha, Faslul Faizi, Aulia Sabri, dan Seluruh sahabat leting 18. Penulis berterima kasih atas motivasi dan semangat kalian semua.

Banda Aceh, 25 Juli 2023

Penulis,

Darul Fata

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

I.1 Latar Belakang

Perkembangan dan evolusi teknologi informasi telah mengubah standar dalam berbagai bidang sesuai dengan tuntutan era modern. Penggunaan internet telah menjadi kebutuhan utama masyarakat untuk mendapatkan informasi secara cepat melalui sistem informasi. Fenomena ini ditandai dengan pertumbuhan signifikan jumlah pengguna layanan internet saat ini.

Dari data yang diungkapkan oleh Hootsuite (we are social, 2022), sebanyak 204.7 juta penduduk Indonesia telah menggunakan internet, mencapai 73.7% dari total populasi Indonesia yang mencapai 227.7 juta. Jumlah pengguna layanan internet terus meningkat setiap tahun, sehingga jumlah informasi yang diperoleh dari internet juga meningkat pesat. Dengan perkembangan sistem informasi yang semakin maju, meningkat pula risiko serangan keamanan dari berbagai teknik ancaman. Oleh karena itu, pengujian keamanan sistem aplikasi berbasis web menjadi sangat penting. Badan Siber dan Sandi Negara (BSSN) bersama Indonesia Honeynet Project (IHP) mencatat ada 20.099.971 serangan siber dalam periode Mei 2021 hingga Mei 2022. Semakin berkembangnya teknologi digital juga membawa dampak negatif, seperti kejahatan dunia maya seperti kebocoran data. Jutaan data pribadi pengguna internet telah bocor melalui situs media sosial dan situs resmi pemerintah di Indonesia sejak tahun 2021 (Mahmud Ashari, 2022).

Pada bidang pendidikan, adanya sistem informasi menjadi sebuah standarisasi untuk menunjang sarana kegiatan akademik. Sistem informasi akademik sudah diterapkan hampir seluruh perguruan tinggi dan universitas di Indonesia, tujuannya ialah untuk mempermudah pengelolaan layanan informasi pada peserta didik, pengajar, dan staf administrasi. Universitas Islam Negeri Ar-Raniry (UIN AR-RANIRY) sebagai salah satu Lembaga Pendidikan perguruan tinggi di Indonesia

memanfaatkan jaringan internet dalam menjalankan aplikasi berbasis web sistem informasi akademik (SIAKAD) yang sebelumnya belum pernah dilakukan pengujian terhadap kerentanan sistemnya. Pentingnya pengujian keamanan SIAKAD terletak pada peran vitalnya dalam mengelola data mahasiswa, dosen, dan staf administrasi serta menyimpan informasi penting terkait pendidikan. SIAKAD menghubungkan seluruh civitas akademik untuk penyampaian informasi. Dalam sistem ini terdapat data sensitif, termasuk informasi pribadi, akademik, dan laporan kehadiran. Pengujian keamanan menjadi krusial untuk mengidentifikasi dan menutup potensi celah keamanan, sehingga dapat mencegah risiko kebocoran data yang berdampak merugikan bagi mahasiswa dan lembaga pendidikan.

Penelitian ini berfokus pada penerapkan standar keamanan OWASP pada aplikasi web SIAKAD UIN Ar-Raniry untuk mengevaluasi kerentanan sistemnya. OWASP, sebuah organisasi yang berfokus pada meningkatkan keamanan perangkat lunak dan aplikasi web, dipilih sebagai metode utama dalam penelitian ini karena keahliannya dalam keamanan aplikasi web, statusnya sebagai standar industri terkemuka, ketersediaan alat uji keamanan, komunitas aktif, dan sumber daya edukasi yang luas. Melalui metode dan alat OWASP, celah keamanan dapat diidentifikasi dan tingkat keamanan aplikasi web dapat ditingkatkan. Proses penelitian ini berfokus pada analisis aktif terhadap aplikasi web untuk menemukan kerentanan dan kelemahan sistem. Hasil pengujian akan mengacu pada OWASP Top 10 2021 sebagai standar untuk mengidentifikasi kerentanan. Penelitian ini akan menghasilkan laporan analisis yang akan digunakan untuk mengevaluasi risiko pada sistem informasi akademik. Oleh karena itu, penelitian ini berjudul "Evaluasi Risiko Celah keamanan Menggunakan Metodologi *Open Web Application Security Project* (OWASP) Pada Aplikasi *Web* Sistem Informasi Akademik UIN Ar-Raniry".

I.2 Rumusan Masalah

1. Apa saja risiko celah keamanan yang ada pada aplikasi *web* Sistem Informasi Akademik (SIAKAD) UIN Ar-raniry?
2. Bagaimana cara untuk mengetahui kondisi dan pengukuran tingkat kerentanan Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry?
3. Bagaimana tingkat keamanan Sistem Informasi Akademik (SIAKAD) menggunakan metode OWASP melalui *penetration testing*?

I.3 Batasan Masalah

1. Penelitian ini mengacu pada Metodologi *Open Web Application Security Project* (OWASP) untuk mengevaluasi tingkat keamanan sistem informasi akademik (SIAKAD)
2. Penelitian dilakukan diluar jaringan UIN Ar-Raniry
3. Penelitian ditujukan pada Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry

I.4 Tujuan Penelitian

1. Mengevaluasi risiko celah keamanan yang ada pada aplikasi *web* Sistem Informasi Akademik (SIAKAD) UIN Ar-raniry
2. Menjabarkan celah serta mengukur tingkat kerentanan Sistem Informasi Akademik (SIAKAD) yang perlu diperbaiki
3. Mengetahui tingkat keamanan Sistem Informasi Akademik (SIAKAD) menggunakan metode OWASP.

I.5 Manfat Penelitian

1. Bagi Penulis, mendapat pengetahuan dan pemahaman tentang ilmu keamanan sistem informasi dalam menganalisa keamanan *web*site
2. Bagi Instansi, sebagai acuan untuk bahan evaluasi keamanan sistem informasi akademik (SIAKAD) serta dapat meningkatkan keamanan sistem informasi tersebut
3. Bagi Universitas, sebagai kontribusi karya ilmiah dalam disiplin ilmu Teknologi Informasi dan sebagai tambahan referensi terhadap penelitian keamanan sistem selanjutnya.

BAB II
LANDASAN TEORI

II.1 Sistem Informasi

Secara umum sistem didefinisikan sebagai kumpulan subsistem baik fisik juga non fisik yang saling berafiliasi dan bekerja secara bersama-sama untuk mencapai suatu tujuan pengolahan data menjadi informasi yang bermanfaat. Informasi adalah kumpulan data atau fakta yang diolah menjadi sesuatu yang berguna bagi penerimanya sehingga dapat digunakan sebagai dasar pengambilan keputusan yang benar.

Sistem informasi merupakan deretan seperangkat komponen dalam perusahaan atau organisasi yang terlibat dalam proses pembuatan dan aliran informasi. Sistem dalam organisasi mengatur kebutuhan pemrosesan transaksi harian, mendukung operasi, aktivitas administratif dan strategis organisasi yang menyediakan laporan yang diperlukan kepada pihak eksternal tertentu. Terdapat sejumlah komponen (manusia, computer, teknologi informasi, dan alur kerja), memiliki sesuatu yang diproses (data menjadi informasi), untuk mencapai tujuan dan sasaran (Andihka, 2021).

Menurut Jogiyanto (2005), komponen-komponen sistem informasi, terdiri dari:

1. Perangkat Keras

   Kumpulan perangkat keras yang dapat membentuk suatu sistem, seperti komputer, printer, dan jaringan.
2. Perangkat Lunak

   Kumpulan perintah/fungsi yang ditulis dengan aturan khusus yang memerintahkan komputer untuk melakukan fungsi tertentu.
3. Data

   Elemen dasar informasi berupa fakta yang membangkitkan kejadian nyata dan menanamkan kedalam simbol.

4. Prosedur

   Suatu fase yang dinyatakan dalam bentuk rangkaian kegiatan yang dihubungkan bersama untuk mencapai suatu tujuan dalam bentuk dokumen prosedural.

5. Manusia

   Pelaku sistem informasi seperti operator, programmer, analis, dan desainer.

II.2 Sistem Informasi Akademik (SIAKAD)

Sistem Informasi Akademik Universitas Islam Negeri Ar-Raniry merupakan sistem pengolahan data yang berkaitan dengan proses pendidikan di Universitas, meliputi data mahasiswa, mata kuliah, data fakultas, data dosen, data nilai, pengelolaan mata kuliah, serta sistem penyimpanan data dan dokumen untuk pendukung pengambilan keputusan oleh pengguna menggunakan sistem komputer. Sistem Informasi Akademik (SIAKAD) menyajikan seluruh konten berupa informasi topik akademik kampus. Selain menjadi sumber informasi kampus, sistem informasi akademik (SIAKAD) berfungsi menjadi alat interaksi antara dosen dengan mahasiswa, mahasiswa dengan mahasiswa, dosen dengan pejabat kampus terkait, dan semua orang dikampus (Al Fathul, 2021).

Jenis-jenis informasi yang terdapat dalam Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry adalah:

1. Berita yang berisi *update* dari institusi Pendidikan dan informasi teknis dari berbagai sumber berita.
2. Pendidikan berisi informasi tentang perkuliahan instansi pendidikan dan perguruan tinggi, seperti mata kuliah, materi perkuliahan, pekerjaan aktual, tugas akhir dan penelitian.
3. Komunitas berisi informasi tentang komunitas institusi pendidikan, yang kemudian dibagikan kepada staf, mahasiswa, alumni, akademisi dan pengumuman.

4. Data pribadi, termasuk informasi mahasiswa seperti Kartu Rencana Studi (KRS) berdasarkan mata kuliah yang diprogramkan selama satu semester.
5. Kartu Hasil Studi (KHS) digunakan untuk mengetahui hasil perkuliahan serta evaluasi pembelajaran dan untuk melihat indeks prestasi belajar mahasiswa.
6. Jadwal handout, yang meliputi jadwal kelas, perencanaan pembelajaran, dan partisipasi dalam perkuliahan.
7. E-mail dapat digunakan sebagai sarana atau alat untuk mengirim dan menerima surat dan pesan.

II.3 Keamanan Sistem Informasi

Menurut Kamus Besar Bahasa Indonesia (KBBI), keamanan diartikan sebagai tidak adanya bahaya. Istilah bahaya disini dapat diartikan sebagai gangguan, ancaman, kecelakaan, atau hal lain yang tidak diinginkan. Dalam konteks keamanan komputer dijelaskan bahwa suatu sistem dapat dikatakan aman jika sumber daya yang digunakan dan aksesnya sesuai dengan keinginan pengguna dalam segala keadaan (Wardana, 2019). Sistem komputer memiliki empat parameter keamanan yang sangat penting, yaitu:

1. *Physical Security* (Keamanan Fisik)

   Keamanan fisik disini menggambarkan perlindungan pertama yang berhubungan langsung dengan perangkat keras. Keamanan ini melindungi dan mencegah hal-hal yang tidak diinginkan pada peralatan komputer.

2. *System Security* (Keamanan Sistem)

   Keamanan sistem membahas perlindungan lebih lanjut seperti bagaimana pengguna dapat mengakses sistem, dan siapa saja yang berhak mengakses sistem.

3. *Application Security* (Keamanan Aplikasi)

Keamanan aplikasi menjelaskan seberapa aman aplikasi digunakan, apakah ada celah dalam aplikasi yang digunakan, dan apakah aplikasi dapat disusupi.

4. *Data Security* (Keamanan Data)

   Keamanan data menggambarkan seberapa aman data yang disimpan, apakah dapat diakses pihak lain tanpa hak akses, dan apakah dapat dihapus, diubah, atau dibocorkan.

Secara umum, keamanan adalah kualitas atau kondisi yang memastikan bahwa sesuatu bebas dari bahaya. Sedangkan informasi adalah data yang telah melalui proses pengolahan sehingga menjadi konteks yang bermanfaat dan berpengaruh penting bagi pengguna tertentu (Yuswandi, 2020).

Keamanan informasi digunakan untuk mendeskripsikan perlindungan aset informasi, meliputi peralatan, fasilitas, dan data komputer dan non-komputer untuk memastikan kerahasiaan, integritas, dan ketersediaan informasi melalui aplikasi, pendidikan, dan teknologi yang akan digunakan sesuai kebijakan teknis yang berlaku. Keamanan informasi memiliki tiga elemen dasar yang harus dikelola, yaitu kerahasiaan informasi dari pihak yang berwenang, integritas informasi untuk menjamin keakuratan dan kelengkapan, dan ketersediaan informasi. (Eka Fuji Astuti, 2019)

Menurut buku Whitman "*Principles of Incident Response and Disaster Recovery"* (Dewanto, 2018), keamanan sistem informasi memiliki beberapa faktor atau ancaman yaitu :

1. Tindakan Kesalahan atau kegagalan manusia (*Acts of human error or failure*): ancaman dikarenakan kesalahan tindakan manusia dimana insiden itu tidak disengaja atau tanpa maksud jahat.
2. Hak atas kekayaan intelektual atau HAKI (*Compromise to Intellectual Property (IP)*) artinya ancaman pelanggaran ketika menggunakan hak kekayaan intelektual seperti hak cipta, rahasia dagang, merek dagang, dan

hak paten. Pelanggaran hak kekayaan intelektual yang paling umum adalah pembajakan aplikasi.

3. Pelanggaran yang disengaja (*Deliberate acts of trespass*): melakukan akses ilegal ke informasi rahasia atau pribadi. Misalnya, peretas menggunakan perangkat lunak untuk mendapatkan akses ilegal ke informasi.
4. Pemerasan: menuntut kompensasi atas pengembalian informasi sensitive yang diterima penyerang.
5. Sabotase atau vandalisme yang disengaja: upaya untuk menghancurkan aset atau merusak citra organisasi.
6. Pencurian yang disengaja: secara illegal mencuri sesuatu milik orang lain.
7. Serangan perangkat lunak: *Software* berbahaya yang dirancang untuk merusak, menghancurkan, atau menolak layanan ke sistem, termasuk virus, *WROM*, *Trojan Horse*, *backdoors*, serangan *Denial of Service (DoS)*, dan *Distributed Denial of Service (DDoS)*.
8. Fenomena alam: peristiwa tidak terduga seperti kebakaran, banjir, gempa bumi, petir, badai, dan letusan gunung berapi.
9. Penyimpangan kualitas layanan oleh penyedia layanan. Seperti produk atau layanan yang telah dihentikan atau tidak dapat berfungsi dengan baik seperti listrik, dan bandwidth jaringan.
10. Kerusakan peralatan atau kegagalan teknis: cacat pada peralatan yang menyebabkan sistem tidak berfungsi, mengakibatkan layanan berkurang atau tidak tersedia.
11. Kesalahan dan kegagalan perangkat lunak: berisi kesalahan dan kondisi tertentu yang belum diuji, ini mungkin merupakan pintasan yang disengaja diciptakan oleh *programmer* untuk beberapa alasan, akan tetapi lupa untuk dihapus.
12. Teknologi dan peralatan usang: infrastruktur yang sudah terlalu lama dan ketinggalan zaman menyebabkan berkurangnya kemampuan sistem sehingga tidak dapat diandalkan dan tidak dapat dipercaya.

Oleh sebab itu untuk mengurangi risiko-risiko yang ada terhadap keamanan suatu informasi dan untuk menjaga proses bisnis tetap berjalan sebagaimana mestinya agar tidak menimbulkan kerugian terhadap individu, perusahaan atau instansi perlu adanya pengujian terhadap suatu sistem atau *web*site untuk mencari celah keamanan yang terdapat dalam sistem atau *web*site tersebut untuk segera dapat dilakukan pencegahan lebih dini terhadap serangan dari orang yang tidak bertanggung jawab sehingga dapat menimbulkan kerugian.

II.4 Penetration Testing

Metode pengujian penetrasi, atau sering disebut "pentest" adalah praktik pengujian keamanan sistem komputer, jaringan, atau aplikasi *web* untuk mengidentifikasi dan mengevaluasi kerentanan keamanan yang dapat dieksploitasi oleh penyerang dengan memberikan tahapan serangan (*attack*) sistem terhadap sistem (Revo et al., 2020). Sistem dievaluasi dengan melakukan simulasi serangan pada sistem atau jaringan untuk mendeteksi celah keamanan yang disebabkan oleh kerentanan sistem atau jaringan, kesalahan konfigurasi, atau kerentanan operasional dalam proses teknis serta potensi pihak yang tidak berwenang untuk dapat akses ke fitur dan data sistem (U.S. Department of the Interior, 2018).

Laporan hasil *penetration testing* memberi pemilik sistem informasi masukan terhadap kerentanan atau celah keamanan sistem yang dapat digunakan untuk mengevaluasi keamanan sistem tersebut, serta memungkinkan untuk menutup kebocoran dan celah sistem lebih awal dalam mengambil Tindakan pencegahan langsung.

Metode untuk *Penetration Testing* terdiri dari *Black-Box Testing*, *Grey-Box Testing*, dan *White-Box Testing* (Irawan et al., 2018).

1. *Black-Box Testing* merupakan metode *pentesting* dimana penguji tidak memiliki hak akses apapun terhadap sistem.

2. *Grey-Box Testing* merupakan metode *pentesting* dimana penguji memposisikan diri sebagai pengguna dengan akses terbatas.
3. *White-Box Testing* merupakan metode *pentesting* dimana penguji memiliki akses penuh ke aplikasi ataupun kode sumber sistem.

Istilah *Penetration Testing* sering disalahartikan sebagai *Vulnerability Analysis*. *Vulnerability Analysis* adalah proses meninjau sistem untuk mengidentifikasi potensi celah keamanan. Sedangkan *Penetration Testing* mensimulasikan serangan peretas pada sistem untuk memastikan adanya celah keamanan.

II.5 Black Box Testing

Dalam *Black box testing*, penguji adalah orang luar yang tidak memiliki pengetahuan tentang sistem atau jaringan yang diuji. Oleh karena itu, penguji harus menemukan semua informasi yang terkait dengan sistem yang sedang dianalisis dan mengidentifikasi serangan dengan metode eksekusi (Hidayatulloh & Saptadiaji, 2021).

Penetration Testing metode *Black Box Testing* merupakan keterampilan dalam menguji sistem aplikasi *web* yang sedang berjalan tanpa mengetahui apa yang terjadi di dalam aplikasi *web* itu sendiri. Seorang penguji berperan sebagai penyerang serta berusaha menemukan dan mengeksploitasi beberapa kerentanan dalam aplikasi *web*.

II.6 OWASP

OWASP (*Open Web Application Security Project*) adalah organisasi/komunitas terbuka yang berfokus pada keamanan aplikasi untuk meningkatkan kesadaran dan mengingatkan semua *developer* bahwa aplikasi berbasis *web* tidak benar-benar aman. OWASP melakukan penelitian dan menyebarluaskan hasilnya untuk meningkatkan kesadaran akan keamanan aplikasi (OWASP, 2022).

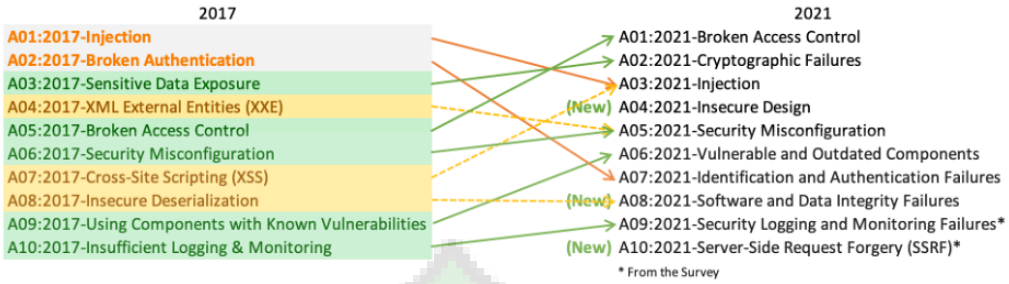
Di OWASP, semuanya gratis dan terbuka untuk pengunjung. Semua alat, dokumentasi, forum, dan cabang OWASP gratis dan terbuka bagi siapa saja yang tertarik untuk meningkatkan aplikasi keamanan. Karena pendekatan keamanan aplikasi yang paling efektif memerlukan peningkatan di semua bidang, OWASP mendukung pendekatan keamanan aplikasi sebagai masalah diskrit, prosedural, dan teknis.

OWASP adalah jenis organisasi yang dapat memberikan informasi keamanan aplikasi dengan cara yang tidak bias, pragmatis, dan hemat biaya. OWASP tidak berafiliasi dengan seluruh perusahaan teknologi. Seperti banyak proyek perangkat lunak *open source*, OWASP membuat berbagai macam konten secara terbuka dan kolaboratif. Yayasan OWASP adalah organisasi nirlaba yang memastikan keberhasilan proyek jangka Panjang. Hampir semua orang yang terlibat dalam OWASP adalah sukarelawan. OWASP ini di bawah skema lisensi Creative Commons Attribution-ShareAlike 2.5. (Keary, 2014).

II.7 OWASP TOP 10

OWASP Top 10 adalah daftar yang diterbitkan oleh komunitas OWASP yang memberikan daftar 10 kerentanan teratas yang dapat mengancam keamanan situs *web*, beserta risiko, dampak dan penyalahgunaannya. Daftar ini terus bertambah dan berubah seiring dengan perkembangan teknologi situs *web* yang terus berkembang. OWASP Top 10 pertama kali diterbitkan pada tahun 2003 dan diperbarui setiap 3 sampai 4 tahun yang tercatat 2003, 2007, 2010, 2013, 2017, dan 2021, daftar kerentanan OWASP terbaru sudah dirilis pada 24 september 2021 (OWASP (Open *Web* Application Security Project), 2021). OWASP Top 10 diurutkan berdasarkan metode kerentanan yang paling sering terjadi.

Perubahan OWASP Top 10 berdasarkan risiko yang paling sering terjadi dari perilisian sebelumnya ke perilisan yang terbaru dapat dilihat pada gambar 2.1.



Gambar 2.1 Perubahan OWASP Top 10 tahun 2017 ke tahun 2021

(Sumber: https://owasp.org/www-project-top-ten/)

Berdasarkan gambar 2.1 diatas, dapat dilihat bahwa daftar kerentanan yang diterbitkan oleh OWASP dapat berubah dan berkembang sesuai dengan *severity* (keparahan) seberapa sering celah keamanan tersebut terjadi dan diperbarui pada perilisan selanjutnya.

Daftar 10 kerentanan pada OWASP Top 10 terbaru yang dirilis pada 24 september 2021:

1. *Broken Access Control* (Kerusakan akses kontrol)

   Kontrol Akses Rusak naik dari risiko paling parah kelima di tahun 2017 menjadi risiko teratas di tahun 2021. Kontrol akses mendefinisikan aturan yang menurutnya pengguna tidak dapat melakukan tindakan apa pun di luar hak akses yang diberikan. Kelalaian untuk melakukannya dapat mengakibatkan pencetakan, modifikasi, atau penghancuran informasi yang tidak sah, atau melakukan aktivitas bisnis di luar batas pengguna.

2. *Cryptographic Failures* (Kegagalan kriptografi)

   Nama yang sebelumnya dikenal sebagai "*Eksposur Data Sensitif* (Paparan Data Sensitif)" telah diubah untuk mencerminkan penyebab masalah dengan lebih baik. Itu naik dari posisi ketiga ke posisi kedua dalam daftar kerentanan OWASP Top 10. Kegagalan kriptografi dengan tidak

melindungi informasi sensitif yang seharusnya tidak berada dalam domain publik.

Kesalahan kriptografi adalah gejala, bukan penyebab. Kesalahan apa pun yang menyebabkan data sensitif dan penting terekspos ke entitas yang tidak sah dapat dianggap sebagai kesalahan kriptografi.

3. *Injection* (Injeksi)

   Injeksi kode terjadi ketika penyerang mengirimkan data palsu ke aplikasi *web* dengan maksud untuk mengelabui agar melakukan sesuatu yang tidak dirancang/diprogram untuk dilakukan oleh aplikasi tersebut.

4. *Insecure Design* (Desain yang tidak aman)

   Merupakan tambahan baru pada daftar OWASP Top 10 berada di nomor empat. *Insecure Design* yang dimaksud berfokus pada pengembangan aplikasi *web* dari awal siklus perancangan.

   Satu dari faktor yang berkontribusi terhadap desain tidak aman adalah kurangnya profil risiko bisnis pada perangkat lunak atau sistem yang akan dikembangkan, sehingga perencanaan tidak dapat menentukan tingkat keamanan informasi yang dibutuhkan.

5. *Security Misconfiguration* (Kesalahan konfigurasi keamanan)

   Kategori ini naik satu tingkat dari daftar OWASP Top10 sebelumnya yang diterbitkan pada tahun 2017. Kesalahan konfigurasi keamanan adalah kode program yang salah dalam sistem aplikasi *web* atau sistem sedang menjalankan proses yang tidak perlu. Ini memungkinkan peretas menyusup ke aplikasi *web* dengan menambahkan virus, dan membuat URL palsu untuk menemukan URL tersembunyi.

6. *Vulnerable and Outdated Components* (Komponen rentan dan kadaluwarsa)

Kategori ini naik dari sebelumnya 2017 di peringkat 9 ke peringkat 6 pada daftar OWSP Top 10 2021. Komponen yang rentan dan usang adalah komponen sistem yang tidak lagi didukung atau telah diidentifikasi rentan terhadap serangan. Seiring pertumbuhan dan perubahan sistem dari waktu ke waktu, komponen seringkali perlu diperbarui atau diganti agar tetap aman. Versi perangkat lunak yang lebih lama biasanya memiliki kerentanan yang telah diketahui sehingga sangat mudah bagi peretas untuk mengeksploitasinya

7. *Identification and Authentication Failures* (Kegagalan identifikasi dan otentikasi)

   Sebelumnya nomor dua dalam daftar OWASP, "*broken authentication* (otentikasi rusak)" telah diubah namanya menjadi ini dan sekarang berada di peringkat tujuh. Celah otentikasi yang rusak dapat memungkinkan penyerang menggunakan metode manual atau otomatis untuk mencoba mengambil kendali atas akun yang diinginkan pada sistem, atau lebih buruk lagi, mendapatkan kendali penuh atas sistem.

   Kegagalan identifikasi dan autentikasi dapat terjadi jika identitas pengguna, autentikasi, atau fitur manajemen sesi belum diterapkan dengan benar, atau jika aplikasi tidak melindunginya secara memadai.

8. *Software and Data Integrity Failures* (Kegagalan integritas data dan perangkat lunak)

   Tambahan baru lainnya pada daftar OWASP Top 10 2021 adalah Kegagalan integritas data dan perangkat lunak. Kesalahan ini dapat terjadi dalam berbagai bentuk, terutama seiring perkembangan *web*, penggunaan kode dan layanan pihak ketiga di situs *web* semakin umum. Contoh kegagalan diantaranya adalah Penggunaan kode yang tidak menjamin integritas sumber, dan Menggunakan plugin pihak ketiga tanpa mengontrol sumbernya

9. *Security Logging and Monitoring Failures* (Kegagalan dalam keamanan login dan monitoring)

   Kategori ini naik satu tingkat dari daftar OWASP Top10 sebelumnya yang diterbitkan pada tahun 2017. Sangat penting untuk memiliki sistem logging dan pemantauan yang berfungsi karena mereka menyediakan log dan informasi yang akan mengingatkan sistem pada waktunya jika terjadi malfungsi atau kegagalan.

   Pencatatan dan pemantauan menjadi sangat penting dalam melacak kembali ketika sistem menunjukkan perilaku abnormal. Kurangnya pencatatan dan pemantauan yang efektif dapat meningkatkan kerusakan situs *web* yang disusupi.

10. *Server-Side Request Forgery (SSRF)* (Pemalsuan permintaan pada server)

    Merupakan tambahan baru pada daftar OWASP Top 10 2021. Kelemahan SSRF terjadi setiap kali aplikasi *web* meminta sumber daya jarak jauh tanpa memvalidasi URL yang disediakan pengguna. Ini memungkinkan penyerang untuk memaksa aplikasi mengirim permintaan ke tujuan yang tidak terduga, bahkan jika mereka dilindungi oleh firewall, VPN, atau beberapa jenis daftar akses jaringan lainnya.

    Ketika layanan cloud meningkat dalam penggunaan dan popularitas serta kompleksitasnya, prevalensi dan risiko serangan SSRF juga meningkat.

II.8 OWASPZap

OWASPZap adalah alat pemindai kerentanan yang dikembangkan oleh organisasi OWASP. Karena alat ini *open source*, siapapun dapat mengembangkan alat ini, sehingga alat ini terus berkembang seiring perkembangan teknologi, itulah sebabnya OWASPZap adalah proyek OWASP yang paling aktif (OWASPZap, 2021).

ZAP (*Zed Attack Proxy*) dari OWASP adalah salah satu alat pemindaian keamanan gratis paling populer di dunia dan dikelola secara aktif oleh ratusan sukarelawan internasional. ZAP dapat secara otomatis memindai kerentanan keamanan dalam aplikasi *web* saat dikembangkan dan diuji. ZAP adalah alat yang andal untuk penguji penetrasi berpengalaman untuk digunakan sebagai alat pengujian keamanan otomatis. ZAP menyediakan *scanner* otomatis sebaik bila kita menggunakan tool untuk menemukan kerentanan secara manual (Zahra, 2020).

ZAP menguntungkan karena strukturnya yang berbasis open source. Meskipun tidak mendeteksi semua kerentanan, program ini dianggap lebih cocok bagi yang baru mengenal penetration testing. OWASPZap memiliki banyak fitur yang membuat pemindaian *web* menjadi lebih mudah. OWASPZap sangat mudah digunakan sehingga memudahkan pemula dalam melakukan scanning terhadap *web* (Ula, 2019).

II.9 Penelitian Relevan

Penelitian relevan adalah sekumpulan sumber penelitian sebelumnya yang telah dikumpulkan dan diterbitkan untuk mendukung penelitian, mendukung referensi penulis dalam penelitian, dan menghindari kesamaan hasil penelitian. Selain itu, penelitian relevan merupakan sumber inspirasi bagi karya tulis penelitian yang ditulis untuk menghasilkan penemuan baru atau untuk dimutakhirkan dengan menggunakan berbagai metode dan jalur ilmiah.

Pertama, penelitian yang dilakukan oleh Rahadiyan Danar Aji tahun 2016 yang berjudul "Evaluasi Risiko Celah Keamanan Menggunakan Metodologi Open *Web* Application Security Project (OWASP) Pada Aplikasi *Web* Sistem Informasi Mahasiswa (Studi Kasus: Perguruan Tinggi XYZ)". Penelitian ini menguji keamanan sistem Aplikasi *web* Sistem Informasi Mahasiswa XYZ (SIMAS-Online) ditemukan memiliki 13 kerentanan risiko. Setelah mengidentifikasi risiko, kerentanan dijelaskan secara lebih rinci dan dianalisis dengan hukum klasifikasi risiko OWASP, yaitu *Risk Rating* (Aji, 2016).

Kedua, penelitian yang dilakukan oleh Adetya Putra Dewanto tahun 2018 yang berjudul "Penetration Testing Pada Domain UII.AC.ID Menggunakan OWASP 10". Penelitian ini menguji keamanan sistem pada 10 target *web* dengan domain uii.ac.id belum memenuhi prinsip keamanan CIA TRIAD, kerahasiaan. Dari hasil *scan* yang dilakukan, ditemukan 10 *web* target menggunakan CPanel dan beberapa celah keamanan yang dapat membahayakan sehingga perlu segera dilakukan Tindakan pencegahan (Dewanto, 2018).

Ketiga, penelitian yang dilakukan oleh Muhammad Subagja Wardana tahun 2019 yang berjudul "Penetration Testing Terhadap *Web*site Asosiasi Pekerja Profesional Informasi Sekolah Indonesia (APISI)". Penelitian ini menguji keamanan sistem situs *web* APISIS memiliki 41 celah keamanan yang dapat dieksploitasi. Dari 41 celah keamanan, 2 dalam kategori critical dan 9 dalam kategori high dan harus segera ditangani. Secara keseluruhan *web*site APISI memiliki skor kerentanan sebesar 6,6 menempatkannya pada kategori medium (Wardana, 2019).

Keempat, penelitian yang dilakukan oleh Al Fathul tahun 2021 yang berjudul "Analisis Tingkat Keamanan Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry Menggunakan Standar ISO 27001;2013 Dengan Klausul 11 dan 14". Penelitian ini menguji keamanan Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry dengan standarisasi ISO 27001;2013 menggunakan klausul 11 dan 14 memiliki rata-rata skor 83,6 dalam persentase (64-73%) maka tingkat keamanan SIAKAD UIN Ar-Raniry dapat dikatakan dalam kategori baik atau cukup aman dari sudut pandang akses operator prodi (Al Fathul, 2021).

Kelima, penelitian yang dilakukan oleh Bella Tasya Kumala Dewi tahun 2022 yang berjudul "*Web Security Compliance to OWASP and SANS Standard*". Penelitian ini menguji keamanan sistem menurut pemetaan yang dibuat berdasarkan daftar kerentanan yang diterbitkan oleh OWASP dan SANS/CWE, situs *web* Cek-ejaan.com memiliki *control akses* yang rusak, kelemahan kriptografi, desain yang tidak aman, dan kegagalan identifikasi dan otentikasi (Dewi, 2022).

Berikut adalah rangkuman penelitian relevan yang terdapat pada tabel 2.1.

Tabel 2.1 Penelitian Relevan

No	Peneliti/Tahun	Judul Penelitian	Isi Penelitian	Hasil Penelitian
1	Rahadiyan Danar Aji (tahun 2016)	Evaluasi Risiko Celah Keamanan Menggunakan Metodologi *Open Web Application Security Project* (OWASP) Pada Aplikasi *Web* Sistem Informasi Mahasiswa (Studi Kasus : Perguruan Tinggi XYZ)	Teknik & Perangkat Pengujian Keamanan Sistem Informasi	Aplikasi *web* Sistem Informasi Mahasiswa XYZ (SIMAS-Online) ditemukan memiliki 13 kerentanan risiko. Setelah mengidentifikasi risiko, kerentanan dijelaskan secara lebih rinci dan dianalisis dengan hukum klasifikasi risiko OWASP, yaitu *Risk Rating*
2	Adetya Putra Dewanto (tahun 2018)	*Penetration Testing* Pada Domain UII.AC.ID Menggunakan OWASP 10	Teknik & Perangkat Pengujian Keamanan Sistem Informasi	Keamanan sistem pada 10 target *web* dengan domain uii.ac.id belum memenuhi prinsip keamanan CIA TRIAD, kerahasiaan. Dari hasil *scan* yang dilakukan, ditemukan 10 *web* target menggunakan CPanel dan beberapa celah keamanan yang dapat

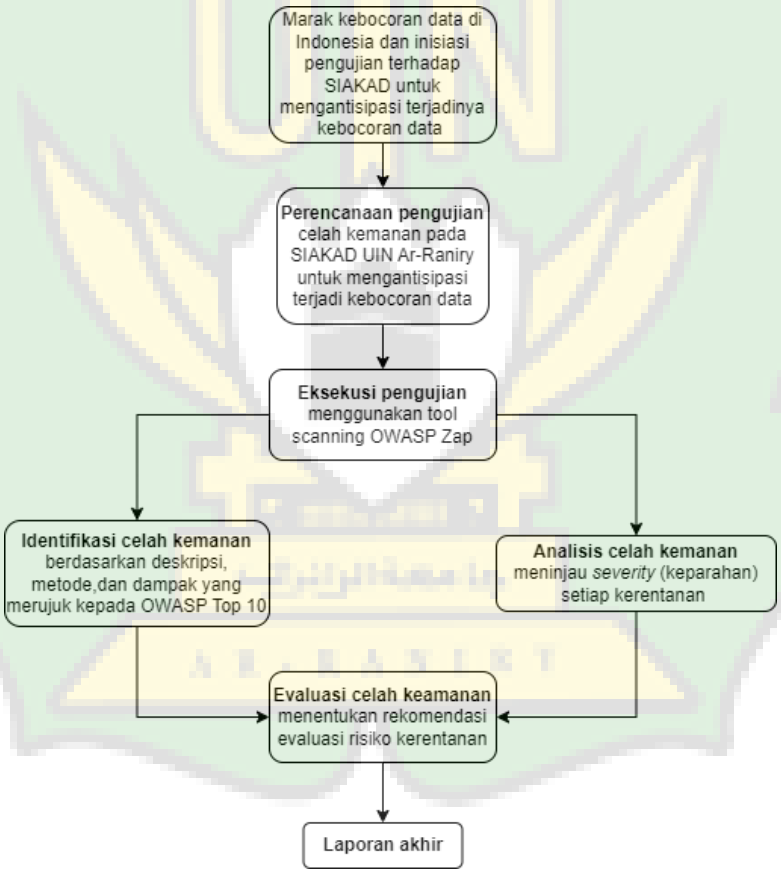
				membahayakan sehingga perlu segera dilakukan Tindakan pencegahan
3	Muhammad Subagja Wardana (tahun 2019)	*Penetration Testing* Terhadap *Website* Asosiasi Pekerja Profesional Informasi Sekolah Indonesia (APISI)	Teknik & Perangkat Pengujian Keamanan Sistem Informasi	Situs *web* APISIS memiliki 41 celah keamanan yang dapat dieksploitasi. Dari 41 celah keamanan, 2 dalam kategori critical dan 9 dalam kategori high dan harus segera ditangani. Secara keseluruhan *web*site APISI memiliki skor kerentanan sebesar 6,6 menempatkannya pada kategori medium
4	Al Fathul (tahun 2021)	Analisis Tingkat Keamanan Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry Menggunakan Standar ISO 27001;2013 Dengan Klausul 11 dan 14	Teknik & Perangkat Pengujian Keamanan Sistem Informasi	Tingkat keamanan Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry dengan ISO 27001;2013 menggunakan klausul 11 dan 14 memiliki skor 83,6 dengan persentase (64-73%) maka tingkat keamanan SIAKAD

				UIN Ar-Raniry dapat dikatakan dalam kategori baik atau cukup aman dari sudut pandang akses operator prodi
5	Bella Tasya Kumala Dewi (tahun 2022)	*Web Security Compliance to OWASP and SANS Standard*	Teknik & Perangkat Pengujian Keamanan Sistem Informasi	Menurut pemetaan yang dibuat berdasarkan daftar kerentanan yang diterbitkan oleh OWASP dan SANS/CWE, situs *web* Cek-ejaan.com memiliki control akses yang rusak, kelemahan kriptografi, desain yang tidak aman, dan kegagalan identifikasi dan otentikasi

II.10 Kerangka Teoritis

Penelitian ini dilatarbelakangi oleh meningkatnya kebocoran data yang terjadi di Indonesia (Badan Siber dan Sandi Negara (BSSN), 2022). Untuk mengantisipasi potensi kebocoran data pada sistem informasi akademik yang menyimpan data pribadi mahasiswa dan civitas kampus lainnya, dilakukan perencanaan pengujian sistem tersebut. Eksekusi pengujian menggunakan metode OWASP dengan memanfaatkan alat OWASP Zap. Hasil eksekusi akan diidentifikasi dan diklasifikasikan berdasarkan deskripsi, risiko, dan dampak celah keamanan. Selanjutnya, dilakukan analisis tingkat keparahan berdasarkan standarisasi OWASP Top 10 dan CIA TRIAD untuk menghasilkan rekomendasi tindakan mitigasi untuk mengevaluasi risiko celah keamanan.

Kerangka teoritis dituangkan dalam gambar 2.2.



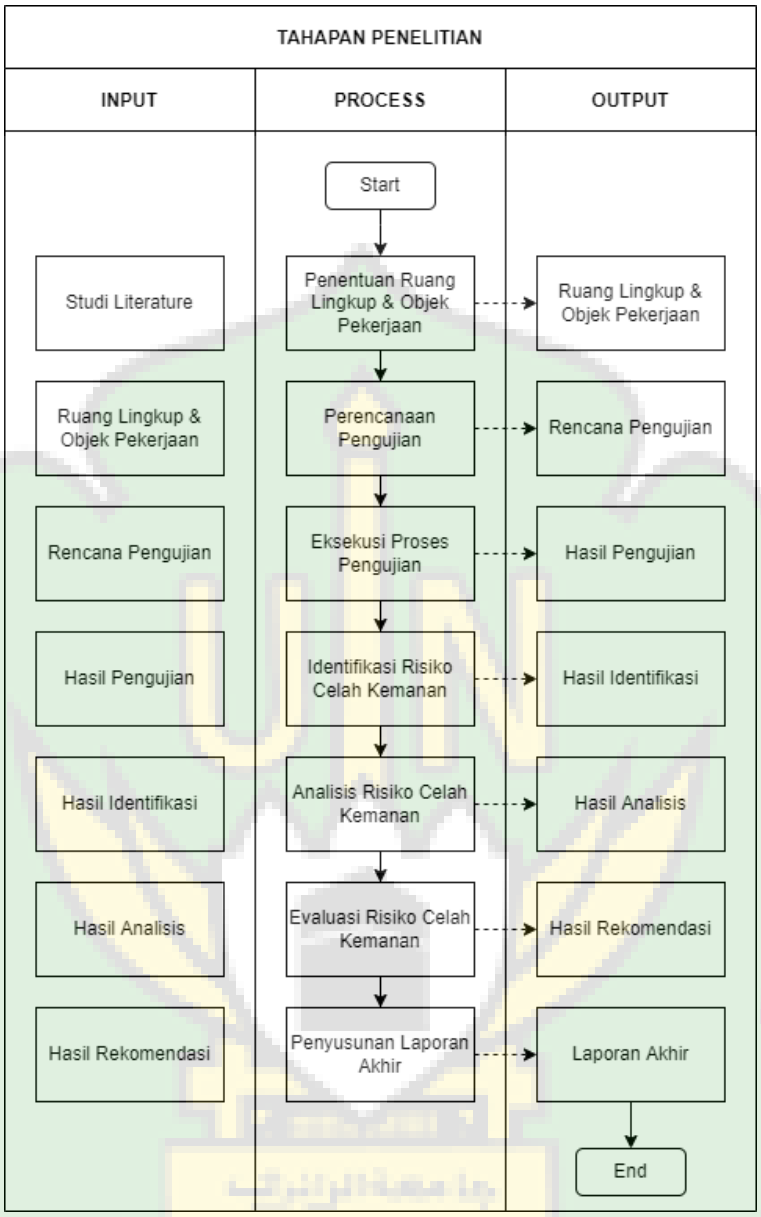
Gambar 2.2 Kerangka Teoritis

BAB III
METODOLOGI PENELITIAN

III.1 Tahapan Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini adalah *Open Web Application Security Project (OWASP)* yang bertujuan untuk menguji keamanan sistem dengan melakukan *penetration testing* dengan menggunakan model *Black-Box Testing* pada Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry. Evaluasi celah keamanan dengan metode OWASP merupakan metode penelitian kualitatif karena lebih fokus pada pemahaman mendalam tentang celah keamanan, eksplorasi potensi risiko, dan deskripsi detail dari hasil penelitian. Penetration testing dilakukan dengan aplikasi otomatisasi OWASPZap. Kemudian dari hasil pengujian akan dilakukan identifikasi dan analisis agar dapat merumuskan evaluasi risiko celah keamanan pada sistem yang mengacu pada standarisasi OWASP TOP 10.

Pada tahapan penelitian ini memberikan gambaran tentang metode kerja yang akan dilakukan untuk menyelesaikan penelitian tugas akhir. Tahapan ini terstruktur dan sistematis karena berfungsi sebagai acuan untuk pengerjaan tugas akhir. Tahapan-tahapan tersebut dapat dilihat pada gambar 3.1.



Gambar 3.1 Tahapan Penelitian

1. **Penentuan Ruang Lingkup & Objek Pekerjaan**

   Menentukan ruang lingkup dan target sistem aplikasi yang akan diuji dengan *penetration testing*. Pada penelitian ini ruang lingkup dan target sistem yang akan diuji adalah Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry.

2. **Perencanaan Pengujian**

   Menyiapkan dan mengkonfigurasi tool OWASP Zap selaku perangkat sistem aplikasi utama dalam eksekusi pengujian, memutuskan alat mana yang akan digunakan untuk melakukan pemeriksaan awal struktur sistem aplikasi menggunakan *footprinting.*

3. **Eksekusi Proses Pengujian**

   Pengujian yang akan dilakukan berfungsi sebagai masukan untuk menilai risiko celah keamanan. Setiap pengujian yang menghasilkan kerentanan akan diidentifikasi, dianalisis, dan ditentukan rekomendasi evaluasi risiko celah keamanannya.

4. **Identifikasi Celah Keamanan**

   Setiap kerentanan yang ditemukan diidentifikasi berdasarkan deskripsi celah keamanan, risiko celah keamanan, dan dampaknya terhadap aplikasi. Segala sesuatu yang ditimbulkan oleh kerentanan yang ditemukan akan dijabarkan menjadi bahan pertimbangan ketika menganalisis celah keamanan.

5. **Analisis Celah Keamanan**

   Berdasarkan kriteria OWASP TOP 10, akan ditinjau hasil celah keamanan, id common weakness enumeration (CWE) dan langkah selanjutnya adalah menentukan *severity* (keparahan) setiap kerentanan yang didapat serta dengan penjelasan dari masing-masing celah keamanan terhadap nilai CIA Triad.

6. **Evaluasi Celah Keamanan**

   Langkah terakhir setelah menganalisis kerentanan adalah menentukan rekomendasi evaluasi risiko kerentanan apa harus dilakukan. Rekomendasi Tindakan diberikan untuk meminimalkan celah keamanan yang dihasilkan.

Nilai severity (keparahan) yang dihasilkan untuk setiap kerentanan kemudian diurutkan untuk menemukan prioritas yang harus diperbaiki terlebih dahulu.

7. **Penyusunan Laporan Akhir**

   Melaporkan dan Menyusun hasil evaluasi risiko celah keamanan sistem dalam bentuk dokumen hasil akhir.

III.2 Lokasi dan Objek Penelitian

Lokasi penelitian dilakukan di luar jaringan internet Universitas Islam Negeri Ar-Raniry Banda Aceh. Penelitian ini dilakukan di lingkungan di luar universitas, misalnya di lingkungan peneliti atau di tempat lain yang tidak terhubung dengan jaringan internet universitas tersebut. Objek dari penelitian ini adalah aplikasi *web* Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry.

III.3 Alat Penelitian

Alat yang diperlukan untuk melaksanakan penelitian ini terdiri dari perangkat keras dan perangkat lunak. Analisis alat dan kebutuhan sistem yang diperlukan untuk penelitian meliputi:

1. Perangkat Keras

   Spesifikasi perangkat keras yang digunakan dalam pengumpulan data penelitian tercantum pada tabel 3.1 berikut:

Tabel 3.1 Spesifikasi Perangkat Keras yang digunakan

Komponen	Spesifikasi
CPU	*Intel® Core™ i3-4030U Processor (up to 1.90GHz)*

RAM	*10GB DDR3 1600Mhz*
Storage	*500GB HDD (ST500LT012-1DG142)*
Graphic	*Discrete graphics Intel® HD Graphics Family / Nvidia GeForce 820M 2GB*

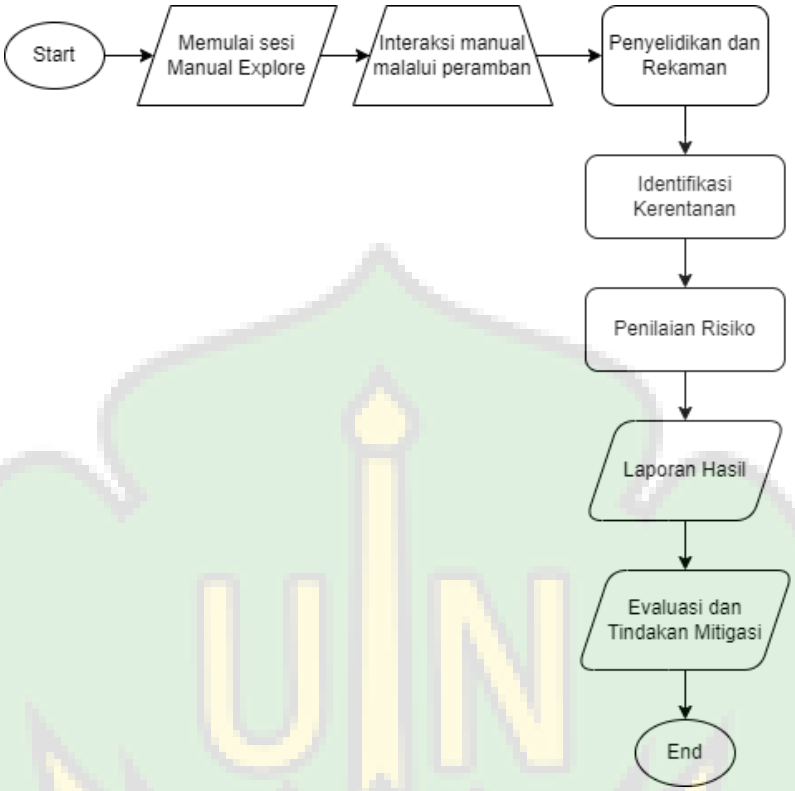
Berdasarkan tabel diatas dapat dilihat spesifikasi *hardware* yang digunakan dalam penelitian ini terdiri dari *CPU (Central Processor Unit)* yang memiliki spesifikasi *Intel® Core™ i3-4030U Processor* dan c*ore speed up to 1.90GHz*, komponen *RAM (Random Access Memory)* sebesar 10GB, komponen penyimpanan *HDD (Hard Disk Drive)* sebesar 500GB, dan komponen *Graphic* dengan spesifikasi *Discrete graphics Intel® HD Graphics Family / Nvidia GeForce 820M 2GB.*

2. Perangkat Lunak
   a. Sistem Operasi Kali Linux versi linux.5.2.0-kali

      Perangkat lunak sistem operasi yang digunakan dalam penelitian ini adalah Kali Linux versi linux.5.2.0-kali. Kali Linux adalah distribusi Linux berbasis Debian yang bersifat *open source* dengan ratusan alat yang dapat digunakan untuk keamanan informasi, seperti *penetration testing*, forensik komputer, rekayasa terbalik, riset keamanan, dan pengujian keamanan tingkat lanjut.

   b. Aplikasi OWASPZap versi 2.13.0

      Aplikasi otomatisasi OWASPZap yang digunakan untuk mendukung proses *penetration testing* memiliki alur yang ditunjukkan pada gambar 3.2.



Gambar 3.2 Alur aplikasi OWASPZap dengan *manual explore*

Pada proses *Manual Explore* menggunakan aplikasi OWASP Zap, pengguna melakukan konfigurasi awal dan menentukan URL target aplikasi web yang akan diuji keamanannya. Selama sesi *Manual Explore*, pengguna secara manual berinteraksi dengan aplikasi web melalui *browser*, melakukan input data, mengklik tautan, dan menjalankan aksi lainnya. Selama interaksi ini, OWASP Zap merekam permintaan dan respon yang terjadi di antara *browser* dan *server*. Setelah berinteraksi, OWASP Zap menganalisis rekaman tersebut untuk mengidentifikasi potensi celah keamanan. Setiap kerentanan yang ditemukan akan dinilai risiko dan keparahannya berdasarkan metode OWASP Top 10 dan faktor lain seperti dampak potensial dan kemungkinan eksploitasi. Hasilnya, OWASP Zap menghasilkan laporan berisi daftar kerentanan yang ditemukan beserta tingkat risiko dan

rekomendasi tindakan mitigasi untuk meningkatkan keamanan aplikasi web.

BAB IV
HASIL DAN PEMBAHASAN

IV.1 Waktu dan Ruang Lingkup

Waktu pengujian dilaksanakan dalam rentang tanggal 5 April 2023 sampai dengan 10 April 2023 dengan cakupan hari kerja Senin sampai dengan Jumat. Hal ini bertujuan untuk pengambilan sampel yang dieksekusi pada hari Senin sebagai hari kerja pertama yang dianggap sebagai hari dengan *traffic* penggunaan *website* paling tinggi, hari rabu sebagai pertengahan minggu dalam hari kerja, dan hari jumat sebagai hari terakhir dalam hari kerja.

Pengujian dilakukan pada saat jam kerja yaitu rentang waktu pukul 08.00 sampai dengan 17.00 pada setiap hari yang telah ditentukan. Hal ini bertujuan sebagai simulasi *attacker* yang sesungguhnya. Adapun ruang lingkup pada pengujian ini adalah Aplikasi *Web* Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry.

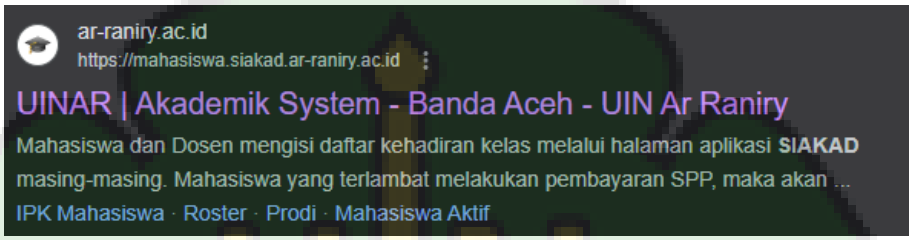
IV.2 Perencanaan Pengujian

Pada tahap ini menjelaskan rencana pengujian penilaian risiko celah keamanan, yang memandu langkah-langkah yang nantinya akan disiapkan sebelum melakukan pengujian kerentanan.

IV.2.1 *Information Gathering*

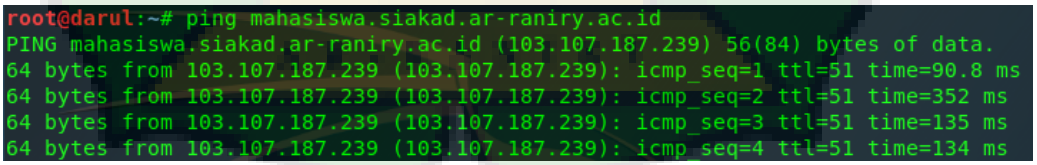
Sebelum melakukan eksekusi pengujian diperlukan berbagai macam informasi mengenai Aplikasi *Web* Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry. Informasi tersebut akan digunakan untuk membantu melakukan pemeriksaan awal struktur sistem aplikasi menggunakan *footprinting.* Setiap temuan informasi kemudian di dokumentasikan dalam bagian pengumpulan data.

Pada tahap pengumpulan data, beberapa alat *footprinting* digunakan di sini melalui Kali Linux dan mesin pencari seperti Google untuk mendapatkan informasi yang berguna untuk melakukan tes penetrasi, seperti sistem operasi yang digunakan, layanan yang berjalan pada setiap sistem, kontak pribadi karyawan yang mengelola SIAKAD UIN, yang semua informasinya dapat didokumentasikan dalam bagian ini.

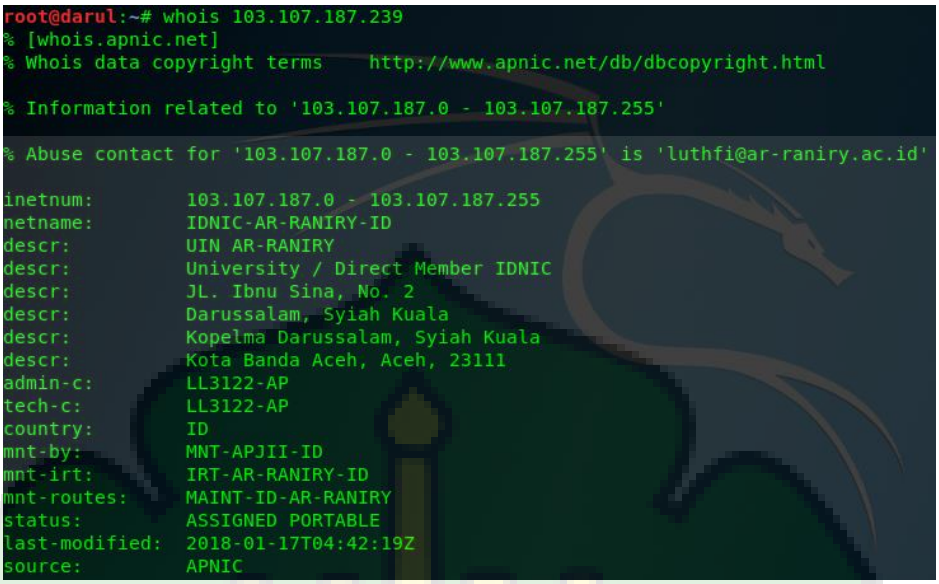


Gambar 4.1 Domain *web* SIAKAD UIN Ar-Raniry

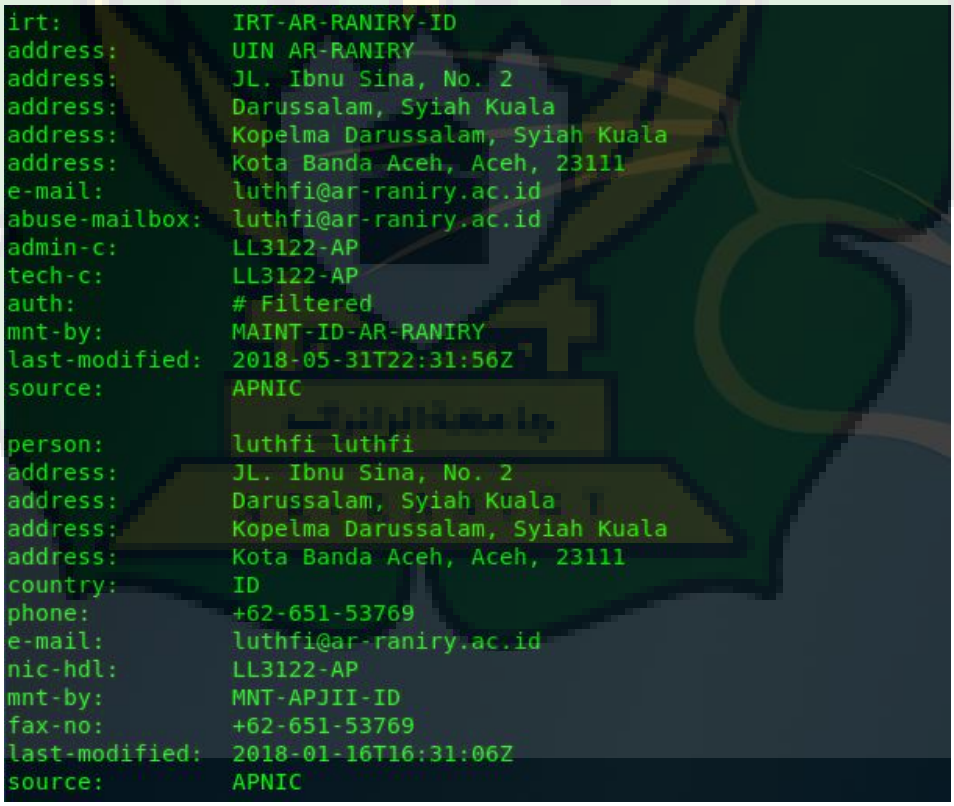
Gambar 4.1 menunjukkan bahwa *web* sistem informasi akademik (SIAKAD) UIN Ar-Raniry memiliki domain mahasiswa.siakad.ar-raniry.ac.id. setelah mengetahui domain untuk mendapatkan informasi berupa ip address dari *web* tersebut dengan menggunakan perintah ping seperti yang terlihat pada gambar 4.2 dimana setelah menjalankan perintah ping terdapat informasi bahwa situs mahasiswa.siakad.ar-raniry.ac.id memiliki ip address 103.107.187.239 dan untuk mendapatkan informasi yang lebih lengkap lagi mengenai situs tersebut maka digunakan tool whois sehingga mendapatkan informasi yang lebih lengkap seperti yang ditunjukkan pada gambar 4.3 dan gambar 4.4



Gambar 4.2 Ping pada *web* SIAKAD UIN Ar-Raniry



Gambar 4.3 Whois pada *web* SIAKAD UIN Ar-Raniry 1



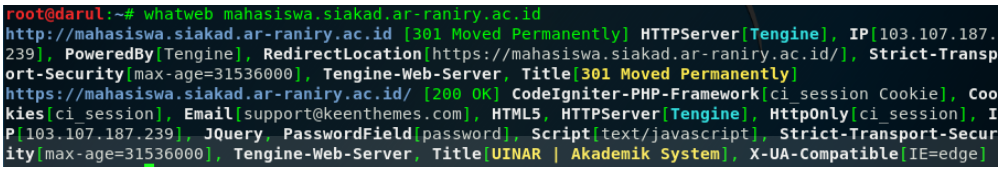
Gambar 4.4 Whois pada *web* SIAKAD UIN Ar-Raniry 2

Berdasarkan informasi yang ditunjukkan pada Gambar 4.3 dan Gambar 4.4, jaringan SIAKAD UIN Ar-Raniry memberikan informasi yang lebih lengkap dan personal seperti nama pegawai, alamat email dan nomor telepon staff pengelola *web* tersebut yang dimana informasi seperti ini dapat digunakan untuk melakukan jenis serangan lain yaitu rekayasa sosial, dimana penyerang biasanya menggunakan informasi yang diperoleh untuk menipu korban, baik melalui *phishing* atau teknik rekayasa sosial lainnya. Kemudian gunakan dnsrecon untuk mengumpulkan informasi tentang server DNS yang digunakan dalam jaringan SIAKAD UIN Ar-Raniry.

Gambar 4.5 Hasil scan DNSRecon

Berdasarkan hasil scan menggunakan dnsrecon yang ditunjukkan pada gambar 4.5 terdapat nameserver milik *web* SIAKAD UIN Ar-Raniry yaitu ns1.rumah*web*.com. Selain itu, ada juga bukti bahwa DNSSEC tidak dikonfigurasi sehingga berpotensi terjadinya dns spoofing dimana hal tersebut dapat mengakibatkan attacker mengalihkan domain atau ip address SIAKAD ke server attacker untuk tujuan tertentu.

Setelah menerima informasi tentang server DNS yang digunakan, selanjutnya mengumpulkan informasi tentang sistem dan alat yang digunakan untuk membuat situs *web* dengan What*Web*.



Gambar 4.6 Hasil scan dengan What*web*

Berdasarkan keterangan yang ditunjukkan pada gambar 4.6 terdapat informasi tambahan yaitu *web* SIAKAD menggunakan *web* server tengine.

Hasil dari proses information gathering yang dilakukan agar mendapatkan informasi yang berkaitan dengan target yang diinginkan antara lain:

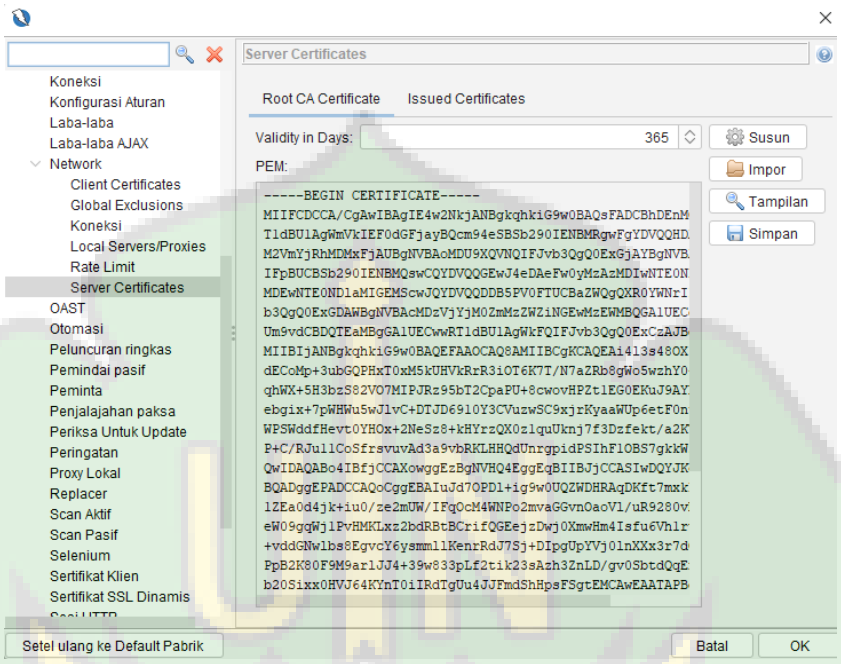
a. *Web* sistem informasi akademik (SIAKAD) UIN Ar-Raniry memiliki domain mahasiswa.siakad.ar-raniry.ac.id dengan ip address 103.107.187.239
b. Memiliki name server yaitu ns1.rumah*web*.com.
c. Mendapatkan beberapa informasi personal pengelola sistem
d. Tidak mengkonfigurasi DNSSEC sehingga berpotensi mengakibatkan terjadinya dns spoofing
e. Menggunakan *web* server tengine.

IV.2.2 Konfigurasi Aplikasi OWASP Zap

Tahap ini perlu dilakukan sebelum pengujian agar aplikasi OWASPZap dapat beroperasi dengan baik terhadap peramban. Proses konfigurasi bertujuan untuk menghubungkan aplikasi dengan peramban terhadap lalu lintas *request* dan *respons* yang diberikan. Langkah awal adalah dengan mengunduh *root ca certificate* pada owaspzap untuk di import dalam peramban yang ingin di sinkronisasi dengan aplikasi owaspzap.

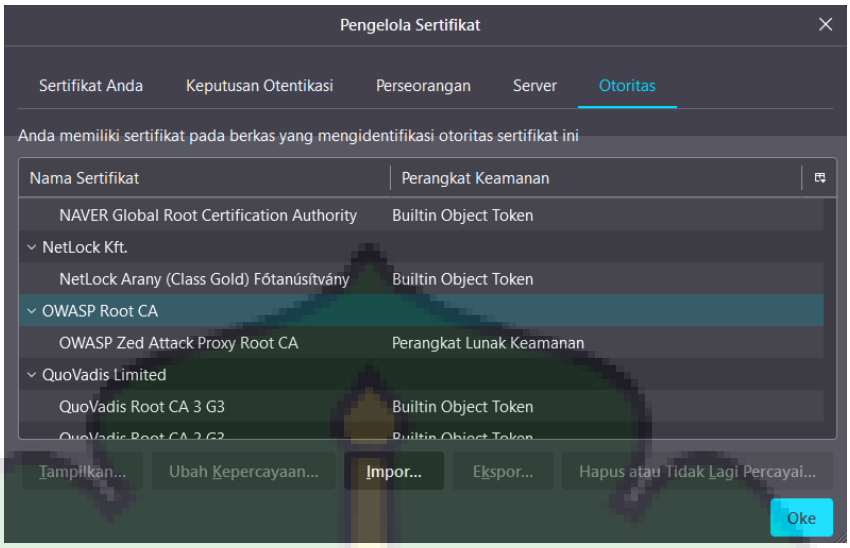
OWASP Zap Root CA certificate merupakan sebuah sertifikat otoritas (Certificate Authority) yang menjadi bagian dari fitur proxy yang ada dalam OWASP Zap. Dalam penggunaannya, ketika fitur proxy diaktifkan pada

OWASP Zap, perangkat lunak ini berperan sebagai perantara antara peramban (browser) pengguna dan server *web* yang sedang diuji.



Gambar 4.7 Root CA Certificate pada aplikasi OWASPZap

Pada gambar 4.7 menunjukkan bagian dari *root ca certificate* yang akan di unduh. Berkas tersebut dapat diakses melalui bilah menu opsi pada *toolbar tools* dalam aplikasi OWASPZap yang telah di instal. Setelah mengunduh berkas langkah selanjutnya adalah mengimpor berkas ke dalam *certificate manager* pada peramban. Dalam kasus pengujian ini menggunakan mozila firefox sebagai peramban untuk disinkronkan dengan aplikasi.



Gambar 4.8 OWASP Root CA berhasil di *import*

Sertifikat SSL/TLS adalah objek digital yang memungkinkan satu sistem memverifikasi identitas dan kemudian membuat koneksi jaringan terenkripsi ke sistem lain menggunakan protokol *secure sockets layer/transport layer security*. Pada mozila firefox menu certificate manager dapat diakses melalui menu pengaturan pada bilah privasi dan keamanan di bagian sertifikat. Terlihat pada gambar 4.8 *root ca certificate* OWASP berhasil di *import*. Langkah ini diperlukan agar dapat menavigasikan apa yang diakses melalui browser dan melihat pesan yang dikirim dan yang diterima terhubung dengan aplikasi.

Langkah selanjutnya adalah dengan mengatur pengaturan koneksi agar lalu lintas *web proxy* terhubung melalui OWASPZap hal ini dapat dilakukan dengan mengubah *proxy* pada *network setting* di peramban untuk menggunakan proxy dan *port* yang sama dengan aplikasi OWASPZap. OWASPZap menggunakan *proxy localhost* yaitu 127.0. 0.1 dengan port 8081 pengaturan ini dapat disesuaikan melalui OWASPZap yang digunakan.

IV.3 Eksekusi Proses Pengujian

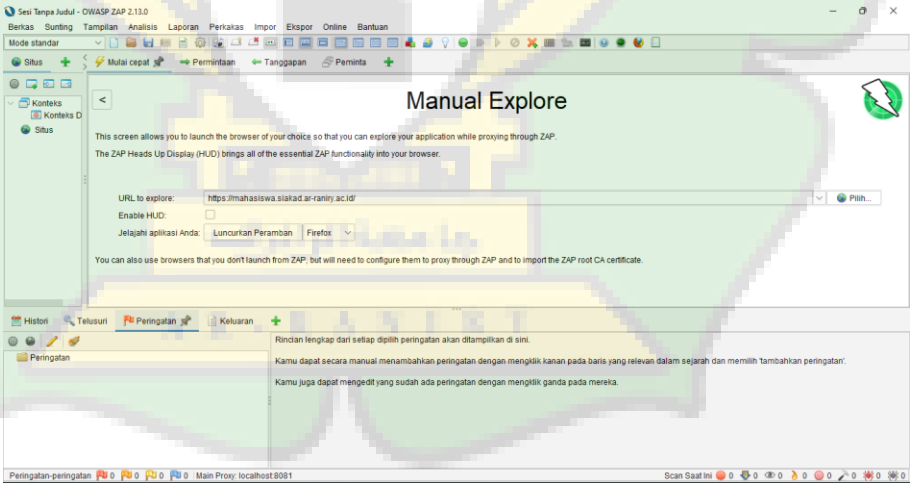
Eksekusi proses pengujian menggunakan OWASPZap 2.13.0. Eksekusi proses pengujian akan dilakukan dengan *manual explore* untuk memperoleh hasil yang lebih terfokus dan tepat sasaran.

IV.3.1 *Manual Explore*

Melalui pendekatan *manual explore*, pengujian dapat dilakukan dengan berinteraksi langsung pada aplikasi *web* yang sedang diuji. Pengguna dapat mengirim permintaan langsung, mengeksplorasi beragam fitur dan fungsionalitas, serta menganalisis respon yang diterima dari server secara langsung.

Langkah dalam melakukan proses manual explore pada OWASP ZAP sebagai berikut:

1. Pada halaman utama aplikasi OWASP ZAP, pilih manual explore kemudian masukkan URL target serta browser yang akan digunakan untuk melakukan manual explore kemudian klik launch browser untuk memulai proses seperti terlihat pada Gambar 4.10



Gambar 4.9 Eksekusi proses pengujian dengan Manual Explore

2. Tunggu beberapa saat hingga browser akan muncul dan proses manual explore siap dilakukan. OWASP ZAP akan secara

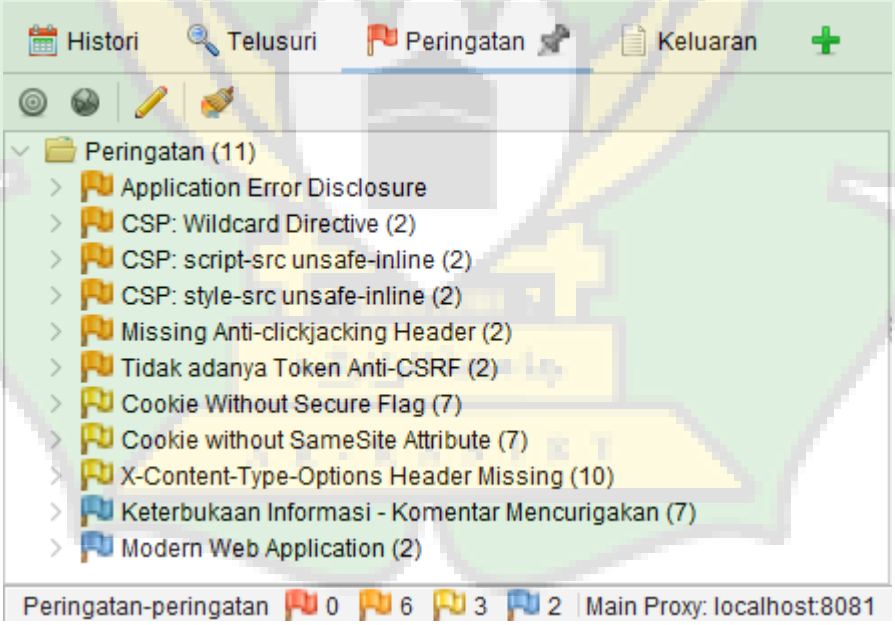
otomatis mendeteksi kerentanan disetiap load atau intruksi yang dijalankan terhadap halaman target.

3. Langkah tersebut juga bisa dilakukan melalui peramban yang sudah mengimport *OWASP root CA sertficate* dan terkonfigurasi *proxy* dan *port* dengan *owasp zap* secara langsung.

IV.4 Identifikasi Celah Keamanan

Setiap celah keamanan yang terdeteksi diidentifikasi berdasarkan penjelasan tentang celah tersebut, risiko yang terkait, dan dampaknya terhadap aplikasi. Segala konsekuensi yang diakibatkan oleh kerentanan yang ditemukan akan diperhitungkan sebagai bahan pertimbangan dalam analisis keamanan.

Berikut adalah hasil dari eksekusi pengujian menggunakan OWASP Zap dengan pendekatan *manual explorer* yang ditunjukkan pada gambar 4.10.



Gambar 4.10 Hasil eksekusi pengujian dengan OWASP Zap

Setelah proses scanning selesai, OWASP Zap akan menghasilkan laporan yang berisi daftar kerentanan yang ditemukan dan tingkat keparahannya. Tingkat

keparahan ditentukan berdasarkan standar OWASP Top 10, yang merupakan daftar sepuluh celah keamanan web paling umum dan berbahaya. Untuk menentukan tingkat keparahan, OWASP Zap mempertimbangkan kompleksitas kerentanan, dampak potensial, kemungkinan eksploitasi, dan tingkat akses yang diperlukan untuk mengeksploitasi kerentanan tersebut. Setiap kerentanan akan diberi tingkat keparahan yang sesuai, yaitu rendah, sedang, atau tinggi, berdasarkan hasil evaluasi tersebut.

Tingkat Risiko	Number of Alerts
Tinggi	0
Sedang	6
Lemah	3
Informasi	2

Gambar 4.11 *Summary of alert* dari hasil eksekusi pengujian

OWASP Zap menentukan risk level (tingkat risiko) kerentanan dengan menggunakan beberapa acuan. Pertama, OWASP Zap mengacu pada daftar OWASP Top 10 yang berisi celah keamanan web paling umum dan berbahaya. Selanjutnya, OWASP Zap mempertimbangkan kompleksitas, dampak potensial, kemungkinan eksploitasi, dan tingkat akses yang diperlukan untuk mengeksploitasi kerentanan. OWASP Zap menggunakan daftar OWASP Top 10 sebagai acuan dalam menentukan risk level karena daftar ini mencakup celah keamanan web paling umum dan berbahaya. Selama proses scanning, OWASP Zap memeriksa apakah aplikasi memiliki kerentanan yang sesuai dengan daftar tersebut. Berdasarkan evaluasi ini, OWASP Zap menetapkan tingkat risiko untuk setiap kerentanan, seperti rendah, sedang, atau tinggi.

Tingkat *High* menunjukkan masalah serius yang berpotensi berdampak besar, *Medium* adalah masalah yang dapat dimanfaatkan dengan kondisi tertentu, *Low* adalah masalah keamanan dengan dampak kecil, dan *Information* adalah informasi berguna tanpa ancaman langsung terhadap keamanan. Pada gambar 4.11 memperlihatkan bahwa hasil dari eksekusi pengujian berdasarkan kategori terdapat 6 macam *alert* yang memiliki level *medium* yang 3 diantaranya merupakan kelompok

yang sama, 3 macam *alert* yang memiliki level *low* dan 2 macam *alert* yang hanya berupa *information.*

Dalam tabel identifikasi celah keamanan, perlu dijelaskan secara rinci tentang celah-celah keamanan yang telah ditemukan. Deskripsi akan bersifat umum dan akan mengacu pada CWE (*Common Weakness Enumeration*). Dari celah keamanan tersebut kemudian dilakukan identifikasi risiko, dan dampak yang ditimbulkan yang akan ditampilkan dalam tabel 4.1.

Tabel 4.1 Deskripsi, Risiko, dan Dampak

Nama Celah	**Deskripsi**	**Risiko**	**Dampak**
Application Error Disclosure	Kerentanan ini mengakibatkan informasi sensitif atau pesan kesalahan teknis dari aplikasi *web* menjadi terbuka bagi pengguna atau penyerang. Situasi ini dapat mengungkapkan rincian konfigurasi, jejak kode, atau bahkan data sensitif. Sebagai hasilnya, penyerang dapat menggunakan informasi ini untuk merancang serangan atau mencari kerentanan	- Pengungkapan informasi *sensitive* - Pemaparan kerentanan aplikasi. - Identifikasi platform dan teknologi yang digunakan.	- Pencurian data sensitif. - Kemungkinan eksploitasi kerentanan. - Gangguan layanan aplikasi. - Kerusakan reputasi bisnis atau organisasi. - Potensi pelanggaran privasi pengguna.

	tambahan.		
CSP (Content Security Policy)	Celah keamanan pada *Content Security Policy (CSP)* terjadi ketika konfigurasi tidak akurat, memberi kesempatan bagi penyerang untuk mengubah kebijakan dan menimbulkan risiko *XSS (Cross Site Scripting)* atau memuat konten dari sumber yang tidak dapat dipercayai. Diperlukan pengaturan dan pengujian CSP yang teliti untuk menghindari celah keamanan ini.	- dapat memungkinkan serangan XSS, di mana penyerang dapat menyisipkan skrip berbahaya ke dalam halaman *web* yang akan diakses oleh pengguna lain. - dapat menyebabkan serangan injeksi data, di mana penyerang dapat memanipulasi data yang ditampilkan di halaman *web* atau menyisipkan konten yang tidak sah.	- Pencurian informasi sensitif. - Perusakan situs. - Penyebaran malware. - Gangguan layanan. - Hilangnya kepercayaan pengguna.
Missing Anti-Clickjacking Header	Celah keamanan "*Missing Anti-Clickjacking Header*" terjadi	- serangan *clickjacking*, di mana pengguna dibujuk untuk	- Pengalihan aksi pengguna. - Potensi penipuan dan *phishing*.

	ketika aplikasi *web* tidak menggunakan *header HTTP seperti "X-Frame-Options"* atau *"Content-Security-Policy"* untuk melindungi dari serangan clickjacking. Hal ini dapat menyebabkan pengguna tanpa sadar mengklik elemen berbahaya di bawah elemen menarik. Penting untuk menyertakan header keamanan yang tepat untuk mencegah serangan *clickjacking*.	melakukan tindakan tanpa sepengetahuan mereka.	- Pengungkapan informasi sensitif. - Merusak reputasi situs *web*.
Tidak adanya *Token* Anti-CSRF(*Cross-Site Request Forgery*)	Celah keamanan "Tidak adanya *Token Anti-CSRF (Cross-Site Request Forgery)*" terjadi saat aplikasi *web* tidak memanfaatkan atau memeriksa token anti-CSRF	- rentan terhadap serangan CSRF, di mana penyerang memaksa pengguna melakukan tindakan yang tidak diinginkan.	- Penyerang dapat memaksa pengguna melakukan tindakan yang tidak mereka sadari - Serangan CSRF dapat digunakan

	dengan benar. Hal ini dapat mengakibatkan serangan CSRF, di mana penyerang memaksa pengguna untuk melakukan tindakan yang tidak diinginkan. Untuk mencegah celah ini, aplikasi harus memastikan penggunaan dan verifikasi token anti-CSRF yang tepat.		untuk mencuri data sensitif dari aplikasi atau pengguna. - merusak reputasi situs *web* jika pengguna merasa tidak aman atau mengalami kerugian akibat aksi tidak diinginkan.
Cookie without Secure Flag	Celah keamanan "*Cookie without Secure Flag*" terjadi ketika cookie yang harusnya dikirimkan secara aman melalui koneksi HTTPS malah dikirimkan melalui koneksi HTTP yang tidak aman.	- Pengiriman cookie melalui koneksi HTTP yang tidak aman.	- Potensi pencurian data sensitif dari cookie. - Kemungkinan eksploitasi oleh penyerang. - Ancaman kerentanan keamanan pada aplikasi *web*.
Cookie without SameSite Attribute	*Cookie* telah diatur tanpa atribut "*SameSite*," yang	- Pengiriman cookie tanpa pengaturan	- Potensi serangan *Cross-Site Request Forgery*

	menyebabkan *cookie* tersebut dapat dikirimkan sebagai hasil permintaan "permintaan lintas situs". Atribut "*SameSite*" berfungsi sebagai langkah pengaman yang efektif untuk mencegah pemalsuan permintaan lintas situs, penyisipan skrip lintas situs, dan serangan berbasis waktu.	atribut "*SameSite*".	(CSRF). - Pencurian data pengguna. - Eksploitasi kerentanan keamanan pada aplikasi *web*. - Ancaman kehilangan kepercayaan pengguna pada situs *web*.
X-Content-Type-Options Header Missing	Celah keamanan "*X-Content-Type-Options Header Missing*" terjadi ketika aplikasi *web* tidak memasukkan atau salah mengonfigurasi header "*X-Content-Type-Options*". Header ini penting untuk mengontrol	- Tidak menyertakan atau tidak mengonfigurasi dengan benar header "*X-Content-Type-Options*" pada respons HTTP.	- Potensi serangan sniffing tipe konten (Content-Type) yang dapat memungkinkan penyerang mengubah interpretasi konten. - Kemungkinan eksploitasi celah keamanan yang

	bagaimana peramban menginterpretasikan jenis konten. Ketika header ini tidak ada, dapat menyebabkan masalah keamanan pada situs *web*.		terkait dengan interpretasi yang salah dari tipe konten.

Risiko mencerminkan sejauh mana kemungkinan suatu celah keamanan atau kerentanan akan dieksploitasi. Fokusnya adalah pada probabilitas terjadinya ancaman keamanan atau potensi kerentanannya. Sementara itu, dampak mencerminkan tingkat kerusakan atau konsekuensi yang dapat terjadi jika celah keamanan atau pelanggaran keamanan terjadi dan berhasil dieksploitasi. Perhatian diberikan pada akibat dari ancaman keamanan yang berhasil dieksploitasi.

IV.5 Hasil Analisis Celah Keamanan

Berdasarkan kriteria OWASP TOP 10, hasil celah keamanan akan ditinjau dan tingkat keparahan masing-masing kerentanan akan ditentukan. Selain itu, kita akan mengevaluasi bagaimana setiap celah keamanan tersebut mempengaruhi nilai-nilai dalam CIA Triad, yaitu Kerahasiaan, Integritas, dan Ketersediaan. CIA Triad adalah konsep keamanan informasi yang menekankan pentingnya menjaga akses terbatas untuk informasi (Kerahasiaan), memastikan informasi akurat dan tidak berubah (Integritas), serta memastikan informasi dapat diakses ketika dibutuhkan (Ketersediaan). CWE *(Common Weakness Enumeration)* merupakan kode khusus yang mengidentifikasi dan mengkategorikan celah keamanan dalam perangkat lunak dan sistem. Standar ini membantu menggambarkan jenis celah keamanan secara konsisten dalam komunitas keamanan siber. Penggunaan ID CWE memfasilitasi komunikasi dan penanganan masalah keamanan secara efisien.

Tabel 4.2 menampilkan korelasi antara celah keamanan dengan OWASP Top 10, CIA Triad, tingkat keparahannya, dan id CWE. Id CWE merupakan identifikasi unik berupa kode untuk setiap jenis celah keamanan, sehingga celah-celah tersebut dapat dibedakan dengan jelas.

Tabel 4.2 Analisis terhadap OWASP Top 10 dan CIA Triad

Nama Celah	**OWASP Top 10**	**CIA TRIAD**	**Risk Level**	**ID CWE**
Application Error Disclosure	A05:2021 – *Security Misconfiguration*	Kerahasiaan (*Confidentiality*): Celah ini mengungkapkan informasi sensitif secara tidak sengaja, mengancam kerahasiaan data aplikasi *web*.	*medium*	200
CSP (Content Security Policy)	A05:2021 – *Security Misconfiguration*	Kerahasiaan (*Confidentiality*): Kesalahan konfigurasi dapat menyebabkan pemblokiran sumber daya yang seharusnya diizinkan atau meningkatkan risiko serangan XSS.	*medium*	693
Missing Anti-Clickjacking Header	A05:2021 – *Security Misconfiguration*	Integritas (*Integrity*): penyerang dapat memanipulasi	*medium*	1021

		tindakan pengguna, mengancam integritas interaksi yang dilakukan oleh pengguna.		
Tidak adanya *Token Anti-CSRF(Cross-Site Request Forgery)*	A01:2021 – *Broken Access Control*	Integritas (*Integrity*): Penyerang dapat memaksa pengguna untuk melakukan tindakan tanpa izin, merusak integritas data dan aksi yang dilakukan oleh pengguna.	*medium*	352
Cookie without Secure Flag	A05:2021 – *Security Misconfiguration*	Kerahasiaan (*Confidentiality*): Jika cookie tidak menggunakan Secure Flag, informasi sensitif dapat dicuri melalui serangan pemantauan jaringan.	*low*	614
Cookie without SameSite Attribute	A01:2021 – *Broken Access Control*	Integritas (*Integrity*): Tanpa SameSite Attribute, cookie dapat digunakan dalam serangan CSRF, mengancam integritas data dan	*low*	1275

		tindakan pengguna.		
X-Content-Type-Options Header Missing	A05:2021 – *Security Misconfiguration*	Integritas (*Integrity*): Jika tidak diatur dengan benar, penyerang dapat memanipulasi tipe konten dan menyebabkan kerentanan keamanan, seperti serangan XSS.	*low*	693

IV.6 Hasil Evaluasi Celah Keamanan

Setelah analisis selesai, langkah selanjutnya adalah melakukan evaluasi celah keamanan. Evaluasi ini melibatkan tindakan mitigasi yang menjadi pertimbangan administrator untuk memperbaiki aplikasi *web*. Tindakan mitigasi adalah upaya untuk mengurangi risiko dan dampak dari celah keamanan atau ancaman. Ini mencakup kebijakan, konfigurasi yang aman, teknologi keamanan, pemantauan, pelatihan, dan pengujian keamanan secara rutin. Mitigasi ini mengacu pada standar CWE (*Common Weakness Enumeration*). Berikut adalah tabel evaluasi celah keamanan:

Tabel 4.3 Evaluasi celah keamanan

Nama Celah	Dampak Terhadap CIA Triad	Tindakan Mitigasi
Application Error Disclosure	Kerahasiaan (*Confidentiality*)	- *Custom Error Handling*: Implementasi pesan kesalahan kustom tanpa informasi sensitif.

		- *Logging* yang Aman: Catat hanya informasi yang diperlukan, hindari data sensitif. - Penyaringan Input: Sanitasi data input pengguna untuk hindari informasi sensitif. - Monitoring Aktivitas: Pantau dan deteksi error secara aktif. - Pengujian Keamanan: Lakukan pengujian rutin untuk identifikasi celah. - Pembaruan dan *Patch*: Selalu pasang *update* keamanan. - Pendidikan Pengguna: Edukasi tentang praktik keamanan.
CSP (Content Security Policy)	Kerahasiaan (*Confidentiality*)	- Terapkan CSP ketat pada server *web*. - Sesuaikan kebijakan sesuai kebutuhan aplikasi. - Pantau dan periksa laporan dari CSP. - Gunakan *mode Report-Only* untuk evaluasi

		awal. - Lakukan pengujian keamanan secara berkala. - Evaluasi laporan untuk perbaiki kebijakan. - Perbarui kebijakan secara rutin.
Missing Anti-Clickjacking Header	Integritas (*Integrity*)	- Gunakan *header X-Frame-Options* dengan nilai "*DENY*" atau "*SAMEORIGIN*". - Implementasi kebijakan CSP untuk mengontrol *framing* konten. - Atur *header X-Content-Type-Options* dengan nilai "*nosniff*". - Pertimbangkan penggunaan *script frame busting.* - Terapkan kebijakan HSTS untuk koneksi HTTPS. - Lakukan pengujian keamanan secara teratur.
Tidak adanya *Token Anti-CSRF(Cross-Site Request Forgery)*	Integritas (*Integrity*)	- Gunakan Token CSRF untuk setiap permintaan modifikasi data. - Periksa *header* HTTP

		untuk token CSRF jika digunakan. - Blokir *Referer header* untuk permintaan merubah data. - Terapkan otorisasi yang kuat. - Hindari menggunakan metode GET untuk aksi penting. - Lakukan pengujian keamanan secara rutin.
Cookie without Secure Flag	Kerahasiaan (*Confidentiality*)	- *Set Secure Flag pada cookie* yang mengandung data sensitif. - Pastikan seluruh situs menggunakan koneksi HTTPS. - Aplikasi server harus menghasilkan *cookie* dengan *Secure Attribute*. - Pemantauan *cookie* secara aktif. - Lakukan pengujian keamanan secara rutin.
Cookie without SameSite Attribute	Integritas (*Integrity*)	- *Set SameSite Attribute* pada *cookie* yang diperlukan. - Pemantauan *cookie*

		secara aktif. - Lakukan pengujian keamanan rutin.
X-Content-Type-Options Header Missing	Integritas (*Integrity*)	- *Set X-Content-Type-Options header* dengan nilai "*nosniff*". - Lakukan pengujian keamanan rutin.

BAB V
Kesimpulan dan Saran

V.1 Kesimpulan

Kesimpulan penelitian berisi hasil analisis dan evaluasi yang telah dilakukan. Berikut adalah beberapa poin penting dari hasil penelitian ini.

1. Setelah melaksanakan pengujian, teridentifikasi sebanyak 9 risiko celah keamanan pada aplikasi *web* Sistem Informasi SIAKAD UIN Ar-Raniry, Terdapat 6 macam *alert* yang memiliki *risk level medium*, dan 3 macam *alert* yang memiliki *risk level low*. yaitu:
   a. *Risk level medium*
   - *Application Error Disclosure*
   - *CSP (Content Security Policy)*
     - *CSP: Wildcard Directive*
     - *CSP: Script-src unsafe-inline*
     - *CSP: Style-src unsafe-inline*
   - Missing Anti-*Clickjacking Header*
   - Tidak adanya *Token Anti-CSRF (Cross-Site Request Forgery)*

   b. *Risk level low*
   - *Cookie without Secure Flag*
   - *Cookie without SameSite Attribute*
   - *X-Content-Type-Options Header Missing*
2. Identifikasi celah keamanan bertujuan untuk memfasilitasi analisis dengan memberikan detail informasi dari setiap celah keamanan yang ditemukan. Identifikasi tersebut meliputi deskripsi umum, risiko, dan dampak dari masing-masing celah keamanan.
3. Analisa risiko celah keamanan melakukan korelasi antara celah keamanan yang telah ditemukan dengan standarisasi OWASP Top 10 dan CIA Triad.
   - Terdapat 2 kategori dari OWASP Top 10 2021
     - A01:2021 – *Broken Access Control*

- A05:2021 – *Security Misconfiguration*

- Terdapat 2 kategori dari CIA Triad
  - Kerahasiaan (*Confidentiality*)
  - Integritas (*Integrity*)

4. Tindakan untuk mengelola dampak yang timbul adalah dengan menerapkan mitigasi sesuai dengan standar CWE (*Common Weakness Enumeration*).
5. Dengan adanya kerentanan sedang dan lemah, potensi risiko bagi keamanan website masih cukup tinggi. Hal ini menunjukkan bahwa ada celah-celah yang dapat dieksploitasi oleh pihak yang tidak bertanggung jawab, dan data serta informasi sensitif dalam website dapat berada dalam risiko terpapar atau digunakan dengan tidak sah.

V.2 Saran

Penelitian ini dilakukan dengan pendekatan *Blackbox Testing*, tetapi beberapa pengujian dalam metodologi *web application penetration testing* dari OWASP memerlukan pendekatan *Whitebox Testing*. Selain itu, untuk mendapatkan pengujian yang lebih mendalam, perlu melakukan pengujian *penetration testing* secara manual, karena pengujian otomatisasi menggunakan OWASP Zap memiliki batasan tertentu.

DAFTAR PUSTAKA

Aji, R. D. (2016). Open *Web* Application Security Project (Owasp) Pada Aplikasi *Web* Sistem Informasi Mahasiswa (Studi Kasus : Perguruan Tinggi Xyz) Vulnerability Risk Evaluation Using Open *Web* Application Security Project (Owasp) Methodology for Student Information. *Open Web Application Security Project (Owasp) Pada Aplikasi Web Sistem Informasi Mahasiswa Perguruan Tinggi Xyz*, 105. http://repository.its.ac.id/71498/1/5212100124-undergraduate-theses.pdf

Al Fathul. (2021). *ANALISIS TINGKAT KEAMANAN SISTEM INFORMASI AKADEMIK (SIAKAD) UIN AR-RANIRY MENGGUNAKAN STANDAR ISO 27001; 2013 DENGAN KLAUSUL 11 DAN 14*.

Andihka, D. A. (2021). *PENGUJIAN KETAHANAN WEBSITE MENGGUNAKAN MODEL PENETRATION TESTING EXECUTION STANDARD (PTES)* (Vol. 3, Issue March).

Badan siber dan sandi negara. (2022). *Informasi Serangan Siber*. https://honeynet.bssn.go.id/

Dewanto, A. P. (2018). Penetration Testing Pada Domain uii.ac.id Menggunakan OWASP 10. In *Https://Dspace.Uii.Ac.Id/*. https://dspace.uii.ac.id/bitstream/handle/123456789/11281/13523025-Adetya Putra D-laporan skripsi.pdf?sequence=1&isAllowed=y

Dewi, B. T. K. (2022). *WEB SECURITY COMPLIANCE TO OWASP AND SANS STANDARD WEB SECURITY COMPLIANCE TO OWASP AND SANS STANDARD*.

Dwi Hastuti. (2016). Sistem Informasi Penomoran Surat (Studi Kasus Fakultas Teknik Universitas Lambung Mangkurat). *Jurnal Teknologi Informasi Universitas Lambung Mangkurat (JTIULM)*, *1*(2), 79–85.

https://doi.org/10.20527/jtiulm.v1i2.11

Eka Fuji Astuti, P. K. S. (2019). ANALISIS BUDAYA KEAMANAN INFORMASI DI KLINIK PRATAMA KOTA BANDUNG. *Jurnal Mitra Manajemen*, *3*(11), 1558–1572. http://e-jurnalmitramanajemen.com/index.php/jmm/article/view/125/69

Ghozali, B., Kusrini, K., & Sudarmawan, S. (2018). Mendeteksi Kerentanan Keamanan Aplikasi *Web*site Menggunakan Metode Owasp (Open *Web* Application Security Project) Untuk Penilaian Risk Rating. *Creative Information Technology Journal*, *4*(4), 264. https://doi.org/10.24076/citec.2017v4i4.119

Guntoro, G., Costaner, L., & Musfawati, M. (2020). Analisis Keamanan *Web* Server Open Journal System (Ojs) Menggunakan Metode Issaf Dan Owasp (Studi Kasus Ojs Universitas Lancang Kuning). *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *5*(1), 45. https://doi.org/10.29100/jipi.v5i1.1565

Hidayatulloh, S., & Saptadiaji, D. (2021). Penetration Testing pada *Web*site Universitas ARS Menggunakan Open *Web* Application Security Project (OWASP). *Jurnal Algoritma*, *18*(1), 77–86. https://doi.org/10.33364/algoritma/v.18-1.827

Irawan, Y., Muzid, S., Susanti, N., & Setiawan, R. (2018). *System Testing using Black Box Testing Equivalence Partitioning (Case Study at Garbage Bank Management Information System on Karya Sentosa)*. 1–7. https://doi.org/10.4108/eai.24-10-2018.2280526

Keary, E. (2014). OWASP - Open *Web* Application Security Project. *OWASP Foundation*, *Cc*, 224. https://www.owasp.org/images/1/19/OTGv4.pdf

Mahmud Ashari. (2022). *No Belajar Dari Kebocoran Data Kredensial: Data Yang Paling Berharga adalah Data Pribadi*. https://www.djkn.kemenkeu.go.id/artikel/baca/14838/Belajar-Dari-Kebocoran-Data-Kredensial-Data-Yang-Paling-Berharga-adalah-Data-Pribadi.html

OWASP. (2022). *About the OWASP Foundation*. https://owasp.org/about/

OWASP (Open *Web* Application Security Project). (2021). *OWASP Top 10 - 2021*. https://owasp.org/Top10/id/A00_2021_Introduction/

OWASPZap. (2021). *Panduan Petunjuk Awal*.

Revo, R., Made, G., Sasmita, A., Agus, I. P., & Pratama, E. (2020). Testing for Information Gathering Using OWASP Testing Guide v4 (Case Study : Udayana University SIMAK-NG Application). *Jurnal Ilmiah Teknologi Dan Komputer*, *1*(1).

U.S. Department of the Interior. (2018). *Penetration Testing*. Office of Chief Information Officer. https://www.doi.gov/ocio/customers/penetration-testing

Ula, M. (2019). Evaluasi Kinerja Software *Web* Penetration Testing. *TECHSI - Jurnal Teknik Informatika*, *11*(3), 336. https://doi.org/10.29103/techsi.v11i3.1996

Wardana, M. S. S. (2019). Penetration Testing Terhadap *Web*site Asosiasi Pekerja Profesional Informasi Sekolah Indonesia. In *Universitas Islam Indonesia*. https://doi.org/.1037//0033-2909.I26.1.78

we are social, H. (2022). *Data (Tren) Pengguna Internet dan Media sosial di Indonesia Tahun 2022*. https://wearesocial.com/uk/blog/2022/04/more-than-5-billion-people-now-use-the-internet/

Yuswandi. (2020). Analisis Kerentanan Keamanan pada Management Information System USAID SEA-PROJECT Menggunakan Metode OWASP. *Jurnal Ilmiah KOMPUTASI*, *19*, 469–482.

Zahra, S. A. Z. (2020). *Analisis Celah Keamanan Pada Laman Login Website Seleksi Elektronik (Online)*.

LAMPIRAN – LAMPIRAN

ZAP Scanning Report

Site: https://mahasiswa.siakad.ar-raniry.ac.id

Summary of Alerts

Tingkat Risiko	Number of Alerts
Tinggi	0
Sedang	6
Lemah	3
Informasi	2

Peringatan

Nama	Tingkat Risiko	Number of Instances
Application Error Disclosure	Sedang	1
CSP: Wildcard Directive	Sedang	2
CSP: script-src unsafe-inline	Sedang	2
CSP: style-src unsafe-inline	Sedang	2
Missing Anti-clickjacking Header	Sedang	2
Tidak adanya Token Anti-CSRF	Sedang	2
Cookie Without Secure Flag	Lemah	7
Cookie without SameSite Attribute	Lemah	7
X-Content-Type-Options Header Missing	Lemah	10
Keterbukaan Informasi - Komentar Mencurigakan	Informasi	7
Modern Web Application	Informasi	2

Alert Detail

Sedang	Application Error Disclosure
Deskripsi	This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation page.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/assets/bundle/mahasiswa/8199.7770143b7262819403d6.js
Metode	GET
Serang	
Evidence	Internal Server Error
Instances	1
Solution	Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user.

Reference	
CWE Id	200
WASC Id	13
Plugin Id	90022

Sedang	CSP: Wildcard Directive
Deskripsi	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	upgrade-insecure-requests
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil
Metode	GET
Serang	
Evidence	upgrade-insecure-requests
Instances	2
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	http://www.w3.org/TR/CSP2/ http://www.w3.org/TR/CSP/ http://caniuse.com/#search=content+security+policy http://content-security-policy.com/ https://github.com/shapesecurity/salvation https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Sedang	CSP: script-src unsafe-inline
Deskripsi	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	upgrade-insecure-requests
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil
Metode	GET
Serang	

Evidence	upgrade-insecure-requests
Instances	2
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	http://www.w3.org/TR/CSP2/ http://www.w3.org/TR/CSP/ http://caniuse.com/#search=content+security+policy http://content-security-policy.com/ https://github.com/shapesecurity/salvation https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Sedang	CSP: style-src unsafe-inline
Deskripsi	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	upgrade-insecure-requests
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil
Metode	GET
Serang	
Evidence	upgrade-insecure-requests
Instances	2
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	http://www.w3.org/TR/CSP2/ http://www.w3.org/TR/CSP/ http://caniuse.com/#search=content+security+policy http://content-security-policy.com/ https://github.com/shapesecurity/salvation https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Sedang	Missing Anti-clickjacking Header
Deskripsi	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	

Evidence	
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil
Metode	GET
Serang	
Evidence	
Instances	2
Solution	Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app. If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
CWE Id	1021
WASC Id	15
Plugin Id	10020

Sedang	Tidak adanya Token Anti-CSRF
Deskripsi	Tidak ada bukti Anti-CSRF yang ditemukan dalam formulir pengiriman HTML. Pemalsuan permintaan lintas-situs merupakan serangan yang melibatkan memaksa korban untuk mengirim permintaan HTTP ke tujuan target tanpa pengetahuan mereka atau niat untuk melakukan suatu tindakan sebagai korban. Penyebab yang mendasari adalah fungsionalitas aplikasi diperiksa menggunakan bentuk/URL tindakan dengan cara berulang. Sifat dari serangan CSRF yang mengeksploitasi kepercayaan bahwa memiliki situs web untuk pengguna. Sebaliknya, lintas-situs penulisan (XSS) mengeksploitasi kepercayaan yang dimiliki pengguna untuk situs web. Seperti XSS, serangan CSRF belum tentu situs-lintas, tapi bisa juga. Permintaan pemalsuan lintas situs juga dikenal sebagai CSRF, XSRF, serangan satu klik, sesi berkuda, deputi bingung, dan ombak laut. Serangan CSRF yang efektif dalam beberapa situasi, termasuk: * Korban telah sesi aktif pada situs target. * Korban yang dikonfirmasi melalui auth HTTP pada situs target. * Korban berada di jaringan lokal yang sama seperti situs target. CSRF terutama telah digunakan untuk melakukan suatu tindakan terhadap situs target dengan menggunakan korban hak-hak istimewa, tetapi beberapa teknik telah ditemukan untuk mengungkapkan informasi dengan meningkatkan akses untuk mendapatkan respon. Risiko pengungkapan informasi secara dramatis meningkat ketika target situs tersebut rentan terhadap XSS, karena XSS dapat digunakan sebagai platform untuk CSRF, yang memungkinkan serangan untuk beroperasi dalam batas-batas kebijakan yang sama-asal.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	<form class="sidebar-search " action="page_general_search_3.html" method="POST">
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil
Metode	GET
Serang	
Evidence	<form class="sidebar-search " action="page_general_search_3.html" method="POST">
Instances	2

Solution	Tahap: Arsitektur dan Desain Diperiksa menggunakan perpustakaan atau kerangka yang tidak memungkinkan kelemahan ini terjadi atau menyediakan konstruksi yang membuat kelemahan ini mudah untuk menghindari. Misalnya, menggunakan paket anti-CSRF seperti OWASP CSRFGuard. Tahap: Implementasi Pastikan aplikasi Anda bebas dari masalah penulisan lintas situs, karena kebanyakan pertahanan CSRF dapat dilewati menggunakan skrip yang dikendalikan oleh penyerang. Fase: Arsitektur dan Desain Hasilkan sebuah unce unik untuk setiap bentuk, letakkan unce ke dalam bentuk, dan verifikasi unce setelah menerima formulir. Pastikan bahwa bukan tidak dapat diprediksi (CWE-330). Perhatikan bahwa ini bisa dilewati dengan menggunakan XSS. Dentifikasi operasi yang sangat berbahaya. Saat pengguna melakukan operasi berbahaya, kirim permintaan konfirmasi terpisah untuk memastikan pengguna berniat melakukan operasi itu. Perhatikan bahwa ini bisa dilewati dengan menggunakan XSS. Gunakan kontrol Manajemen Sesi ESAPI. Kontrol ini mencakup komponen untuk CSRF. Jangan gunakan metode GET untuk setiap permintaan yang memicu perubahan status. Tahap: Implementasi Periksa header Referral HTTP untuk melihat apakah permintaan berasal dari halaman yang diharapkan. Ini bisa melanggar fungsi yang sah, karena pengguna atau proxy mungkin telah menonaktifkan pengiriman Rujukan karena alasan privasi.
Reference	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Plugin Id	10202

Lemah	**Cookie Without Secure Flag**
Deskripsi	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getLinkBidoataPDDikti
Metode	GET
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil/getDetailContent
Metode	GET
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil/hitungAKM

Metode	GET
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getInvoiceSPP
Metode	POST
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getJadwalAbsensiMahasiswa
Metode	POST
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getStatistikContent
Metode	POST
Serang	
Evidence	Set-Cookie: ci_session
Instances	7
Solution	Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.
Reference	https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html
CWE Id	614
WASC Id	13
Plugin Id	10011
Lemah	**Cookie without SameSite Attribute**
Deskripsi	Kuki telah ditetapkan tanpa atribut SameSite, yang berarti cookie tersebut dapat dikirim sebagai hasil permintaan 'permintaan lintas situs'. Atribut SameSite adalah ukuran penghitung yang efektif untuk pemalsuan permintaan lintas situs, penyisipan naskah lintas situs, dan serangan waktu.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getLinkBidoataPDDikti
Metode	GET
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil/getDetailContent
Metode	GET
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil/hitungAKM

Metode	GET
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getInvoiceSPP
Metode	POST
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getJadwalAbsensiMahasiswa
Metode	POST
Serang	
Evidence	Set-Cookie: ci_session
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getStatistikContent
Metode	POST
Serang	
Evidence	Set-Cookie: ci_session
Instances	7
Solution	Memastikan bahwa perlengkapan SameSite diatur baik 'lemah' atau pilihan yang 'ketat' untuk semua cookie.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
CWE Id	1275
WASC Id	13
Plugin Id	10054

Lemah	**X-Content-Type-Options Header Missing**
Deskripsi	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/assets/bundle/mahasiswa/7868.7770143b7262819403d6.js
Metode	GET
Serang	
Evidence	
URL	https://mahasiswa.siakad.ar-raniry.ac.id/assets/bundle/mahasiswa/8199.7770143b7262819403d6.js
Metode	GET
Serang	
Evidence	
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getLinkBidoataPDDikti

Metode	GET
Serang	
Evidence	
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil
Metode	GET
Serang	
Evidence	
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil/getDetailContent
Metode	GET
Serang	
Evidence	
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil/hitungAKM
Metode	GET
Serang	
Evidence	
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getInvoiceSPP
Metode	POST
Serang	
Evidence	
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getJadwalAbsensiMahasiswa
Metode	POST
Serang	
Evidence	
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home/getStatistikContent
Metode	POST
Serang	
Evidence	
Instances	10
Solution	Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application /web server to not perform MIME-sniffing.
Reference	http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx https://owasp.org/www-community/Security_Headers
CWE Id	693
WASC Id	15
Plugin Id	10021

Informasi	**Keterbukaan Informasi - Komentar Mencurigakan**
Deskripsi	The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks or files are against the entire content not only comments.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/assets/bundle/mahasiswa/8199.7770143b7262819403d6.js

Metode	GET
Serang	
Evidence	user
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	username
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil
Metode	GET
Serang	
Evidence	username
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	from
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	USER
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil
Metode	GET
Serang	
Evidence	from
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil
Metode	GET
Serang	
Evidence	USER
Instances	7
Solution	Hapus semua komentar yang mengembalikan informasi yang dapat membantu penyerang dan memperbaiki masalah mendasar yang mereka lihat.
Reference	
CWE Id	200
WASC Id	13
Plugin Id	10027

Informasi	**Modern Web Application**
Deskripsi	The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.
URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/home
Metode	GET
Serang	
Evidence	 Skripsi

URL	https://mahasiswa.siakad.ar-raniry.ac.id/e-mahasiswa/profil
Metode	GET
Serang	
Evidence	 Skripsi
Instances	2
Solution	This is an informational alert and so no changes are required.
Reference	
CWE Id	
WASC Id	
Plugin Id	10109